

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Systém pro bezdrátový sběr a zpracování dat ze senzorů
fyzikálních veličin

System for Wireless Data Acquisition and Processing from Sensors
of Physical Quantities

Zadání diplomové práce

Student: **Bc. Radim Habartík**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2612T003 Aplikovaná elektronika

Téma: **Systém pro bezdrátový sběr a zpracování dat ze senzorů fyzikálních veličin**
System for Wireless Data Acquisition and Processing from Sensors of Physical Quantities

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Proved'te rozbor principů a funkcí vybraných senzorů fyzikálních veličin.
2. Navrhněte systém pro bezdrátový sběr dat s využitím vybraných senzorů.
3. Realizujte navržené zařízení včetně programového vybavení.
4. Funkčnost systému ověřte pomocí měření.

Seznam doporučené odborné literatury:

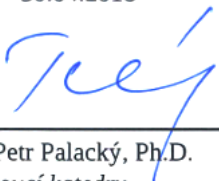
Podle pokynů vedoucího závěrečné práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Martin Sobek, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 25. dubna 2018

Holavský Radim

.....
podpis studenta

Poděkování

Rád bych poděkoval Ing. Martinu Sobkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá rozbořem vybraných sensoru teploty, vlhkosti, tlaku, intenzity světelného záření. Dále pak návrhem a realizací systému sběru dat z těchto vybraných sensorů. Následná implementace sensorů a navrženého systému sběru dat do realizačního projektu s názvem: „MeteoSens“ aneb pevně instalovaná meteostanice. Meteostanice bude sbírat data také z meteorologických aparátů, tj. rychlost a směr větru, srážky. Druhá polovina práce se bude zabývat s nakládáním změřených dat, neboli zobrazení v grafech či zobrazení na displeji. Systém posílání změřených dat bude založen na principu knihoven komunity MySensors, které jsou volně dostupné – OpenSource.

Cílem této diplomové práce je vytvořit funkční zařízení a eliminovat možné faktory, které by způsobily poruchu nebo nestabilitu systému. Zajistit provoz zařízení, dokud se nestane technicky a morálně zastaralým. To bude zajištěno pravidelnou údržbou a aktualizací softwaru.

Klíčová slova

Senzor, Měření fyzikálních veličin, Meteostanice, MySensors, MyControler.org, MQTT

Abstract

This diploma thesis focused on analysis of selected sensors - temperature, humidity, pressure, light intensity. Furthermore, design and implementation of the data collection system from these selected sensors. And implementation of the sensors and the proposed data collection system into the implementation project called "MeteoSens" or a fixedly installed weather station. The weather station will also collect meteorological data, i.e wind speed and direction, rainfall. The second half of the thesis will deal with loading measured data, displaying graphs or show on the display. The system for sending measured data will be based on the principle of MySensors community libraries, which are freely available - OpenSource.

The aim of this diploma thesis is to create functional devices and eliminate possible factors that would cause a system failure or instability. Ensure device operation until it becomes technically and morally obsolete. This will be ensured by regular maintenance and updating software.

Key words

Sensor, Measurement of physical quantities, Weather station, MySensors, MyControler.org, MQTT

Obsah

Seznam použitých symbolů	- 7 -
Seznam použitých zkratk	- 8 -
Seznam ilustrací a seznam tabulek	- 9 -
Úvod	- 12 -
1 Rozbor principů senzorů	- 13 -
1.1 Měření teploty	- 13 -
1.1.1 Kovové odporové teploměry	- 13 -
1.1.2 Termočlánek	- 13 -
1.2 Měření atmosférického tlaku	- 14 -
1.3 Měření intenzity osvětlení	- 15 -
1.4 Měření srážek	- 15 -
1.5 Měření rychlosti a směru větru	- 16 -
2 Popis konkrétních senzorů	- 17 -
2.1 Teploměr SHT31	- 17 -
2.2 Teploměr HTU21	- 18 -
2.3 Tlakoměr BMP280	- 20 -
2.4 Luxmetr BH1750	- 21 -
3 Komunita „MySensors“	- 23 -
3.1 Topologie sítě	- 23 -
3.2 API - Application Programming Interface	- 25 -
3.3 Komunikační rozhraní	- 27 -
3.3.1 Bezdrátový modul nRF24L01+	- 27 -
3.3.2 Ethernet shield s čipem W5100	- 28 -
3.4 The Controller	- 29 -
4 Projekt: „MeteoSens“	- 30 -
4.1 Konstrukční provedení	- 30 -
4.2 DPS jednotky sběru dat	- 31 -
4.3 Principiální schéma systému	- 32 -
4.4 WatchDog	- 33 -
4.5 Software	- 34 -
4.5.1 Použité vývojové prostředí	- 36 -

4.5.2	Debuggovací log.....	- 37 -
4.6	Přijímací jednotka s displejem	- 39 -
5	Ověření funkce a záznamy měření	- 42 -
5.1	Měření teploty.....	- 43 -
5.2	Měření vlhkosti	- 44 -
5.3	Měření atmosférického tlaku.....	- 45 -
5.4	Měření rychlosti větru.....	- 46 -
5.5	Shrnutí výsledků	- 47 -
	Závěr.....	- 48 -
	Použitá literatura	- 50 -
	Seznam příloh	I

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
U	V	Napětí (obecně)
I	A	Proud (obecně)
R_t	Ω	Odpor kovu při dané teplotě
R₀	Ω	Odpor kovu při nulové teplotě
U_{AB}	V	Termonapětí
t	°C	Teplota obecně
RH	%	Relative Humidity (Relativní vlhkost)
P	Pa, hPa	Tlak
E	lx (lux)	Intenzita osvětlení

Seznam použitých zkratk

Zkratka	Význam
μC / MCU	Mikrokontrolér
μPC	Mikropočítač
ADC	Analogově digitální převodník
API	Application Programming Interface
AVR	Označení rodiny 8bitových μC od výrobce Atmel
CPU	Central procesor unit – hlavní procesorová jednotka
DHCP	Dynamic Host Configuration Protocol
DPS	Deska plošných spojů
GW	Brána v síti MySensors
INTx	Vstup pro aktivaci přerušení
LAN	Local Area Network – Lokální počítačová síť
LSB / MSB	Nejméně významný bit / Nejvíce významný bit
MQTT	Message Queuing Telemetry Transport
NODE	Uzel v síti MySensors
NRF24	Označení bezdrátového modulu
RAM	Random Access Memory – Paměť s náhodným přístupem
RS485	Standard sériové komunikace
RX / TX	Přijímání / Odesílání dat
SMD	(SMT) - surface mount technology – povrchová montáž součástek
TTL	Tranzistorově-tranzistorová logika
UART	Universal Synchronous / Asynchronous Receiver and Transmitter
USB	Universal Serial Bus - univerzální sériová sběrnice
WI-FI	Wireless Fidelity – bezdrátová počítačová síť
UTP	Unshielded Twisted Pair
WDT	WatchDog Timer

Seznam ilustrací a seznam tabulek

Číslo ilustrace	Název ilustrace	Číslo stránky
1	3D model tlakoměru od výrobce Bosch	14
2	3D model piézo membrány – deformace materiálu vlivem tlaku	14
3	Srážkoměr ze setu WH1080 + Vnitřní schéma	15
4	Anemometr ze setu WH1080	16
5	Detektor směru ze setu WH1080 + Vnitřní schéma	16
6	čip SHT31	
7	Blok. Schéma SHT31	17
8	Graf přesnosti změřené veličiny v závislosti na teplotě	18
9	3D model čipu HTU21	18
10	Graf přesnosti změřené veličiny v závislosti na teplotě	19
11	Podoba čipu BMP280	20
12	Blokové schéma senzoru BMP280	20
13	Modul BH1750	21
14	Blokové schéma senzoru BH1750	22
15	Logo MySensors	23
16	Nejjednodušší příklad sítě MySensors	24
17	Struktura posílané zprávy (MyMessage)	
18	Originální popis Seriál API, z webu MySensors.org	26
19	Originální popis příkazů (command), z webu MySensors.org	26
20	nRF24	27
21	Eth. modul	28
22	čip W5100	28

23	Uživatelský vzhled aplikace MyController.org	29
24	Podoba vysílací jednotky meteostanice	30
25	Podoba jednotky sběru dat	31
26	Modul Mega-Pro-Mini s procesorem ATmega 2560	31
27	Blokové schéma projektu „MeteoSens“	32
28	Blokové schéma – připojení k serveru (kontroléru)	32
29	Podoba modulu s procesorem ATmega 328p	33
30	Vývojové prostředí Arduino IDE na OS: Raspberrian	36
31	Podoba přijímací jednotky	41
32	Mapa umístění meteostanic, vzdálenost a výškový rozdíl	42
33	Průběh teploty AVG – průměr všech čidel.	43
34	Průběh teploty meteostanice Remak, a.s.	43
35	Průběh relativní vlhkosti, porovnání čidel SHT31,HTU21 zahrada + čidlo HTU21 – krytá terasa	44
36	Průběh relativní vlhkosti meteostanice Remak, a.s.	44
37	Průběh atmo. tlaku, porovnání čidel BMP280-zahrada vs. BMP280-krytá terasa	45
38	Průběh atmo. tlaku meteostanice Remak, a.s.	45
39	Graf rychlosti větru (režim: Min,Max,Avg)	46
40	Výstupní logické úrovně v závislosti na uhlu natočení vrtule	46
41	Průběh napětí na výstupu anemometru	47

Číslo tabulky	Název tabulky	Číslo stránky
1	Základní parametry SHT31	17
2	Základní parametry HTU21	19
3	Základní parametry BMP280	20
4	Klíčové vlastnosti senzoru BH1750	21
5	Základní parametry modulu nRF24L01+	27
6	Klíčové parametry čipu W5100	28
7	Přiřazení ID symbolu z ACII tabulky	40

Úvod

Žijeme v době, kdy se nevyplatí vyrábět si svoje domácí zařízení. Ceny sériově vyráběných produktů jsou natolik nízké, že domácí výroba elektrozařízení se stala minulostí. Existuje už pouze pár oblastí, ve kterých se tato činnost vyplatí. Když pomineme oblast audio zesilovačů, tak druhou velkou oblastí je programování levných mikročipů a implementace svého nápadu. Aplikace s programovým vybavením, které bylo vytvořeno svépomocí, je mnohem lépe přizpůsobitelná a laditelná pro konkrétní podmínky.

Právě proto v posledních 3 letech zažívá „boom“ oblast zvaná „Internet of Thing“ (zkráceně – IOT). Víze tohoto sousloví spočívá v mnoho malých úsporných zařízeních, většinou napájených z baterie, které jsou připojeny do internetu a posílají/přijímají data. Velikost takového zařízení je samozřejmě co nejmenší, např. velikost „krabičky od sirek“. Tato diplomová práce bude mimo jiné pojednávat o upravené verzi IOT neboli tzv. „Internet of your Things“. Původní víze je samozřejmě stejná s tím, že zařízení připojené do internetu nebo pouze do lokální počítačové sítě (LAN), budou mít jeden centrální bod (tzv. Controller-server), který bude od všech sbírat data a může zajistit řízení členů v síti podle určitých podmínek. Nejčastější typy členů v IOT síti: teploměr, vlhkoměr, tlakoměr, luxmetr, relé pro spínání zařízení AC 230V atp.

Tato diplomová práce se bude zabírat rozбором funkce digitálních senzorů, které se nejčastěji používají v sítích IOT, a následné implementace do aplikace „Venkovní meteostanice“. Tato aplikace už se moc projektu IOT podobat nebude, protože jako typický projekt miniaturní velikosti má velký vliv na přesnost měřených veličin a napájení z elektrické sítě zajistí bezstarostný chod oproti napájení z baterie, případně solárních článků. Proto v této aplikaci se nebudu omezovat prostorem ani spotřebou, ale samozřejmě se budu snažit zařízení udělat kompaktní při zachování všech důležitých parametrů.

Pokud se řekne meteostanice, většina z nás si vybaví malou krabičku zakoupenou v elektroprodejně. Taková zařízení většinou lidí postačuje i přes omezené možnosti, absence záznamu dat a omezená vzdálenost mezi stanicí a senzorem, která rapidně klesá s počtem cihlových stěn, kterými musí signál projít. Dále tu máme skupinu lidí, kteří se nazývají amatérští meteorologové, kteří chtějí profesionální stanici, a nebojí se za takové zařízení zaplatit desetitisíce.

A jak už bylo zmíněno výše, takové zařízení se vyplatí postavit svépomocí, protože jednotlivé senzory, radiové moduly, mikro-kontroléry jsou v „čínských e-shopech“ levné položky, a výsledná cena meteostanice je velmi přijatelná, přičemž svojí kvalitou se přiblíží profesionální meteostanici, podle toho kolik na ní konstruktér zanechal práce. Přesnost měření např. teploty samozřejmě závisí na typu použitého senzoru, kdy v cenové relaci hotových meteostanic do 10 000 Kč, jsou použity naprosto identická čidla, nejčastěji tedy teploměr SHT31, který bude popisován níže.

První kapitola obsahuje rozbor senzorů fyzikálních veličin. V druhé kapitole se nachází konkrétní popis vybraných senzorů. Ve třetí kapitole se nachází popis systému přenášených dat (protokoly/standarty komunity „MySensors“). Ve čtvrté kapitole se nachází popis realizovaného zařízení. V páté kapitole se nachází ověření funkce a závěrečná měření.

1 Rozbor principů senzorů

1.1 Měření teploty

Měření teploty neboli Termometrie se posuzuje objektivní míra tepelného stavu látky. Dle jednotek SI je základní veličinou termodynamická teplota, jejíž jednotkou je Kelvin. Teplota se dá měřit pouze nepřímo na základě známých fyzikálních jevů za různých teplot, založenými většinou na objemové roztažnosti kapalin nebo délkové roztažnosti pevných látek za různé teploty. [3]

1.1.1 Kovové odporové teploměry

Princip odporového teploměru je založen na změně elektrického odporu kovu se změnou teploty. Jako nejpoužívanější materiál je Platina, která svými vlastnostmi umožňuje měřit v rozsahu -100 °C až 440 °C.

Klíčové vlastnosti:

- Linearita ve velkém rozsahu teplot,
- snadné vyhodnocení,
- přesnost dle použitého materiálu,
- používá se pro velmi přesná měření v laboratorním zařízení.

Závislost elektrického odporu a teploty udává vztah:

$$R_t = R_0 \cdot (1 + \alpha \cdot t) \quad (1.1)$$

R_t - odpor při teplotě t ,

R_0 - odpor při nulové teplotě

α (K⁻¹) - teplotní součinitel odporu.

1.1.2 Termočlánek

Princip termočlánu je založen na termoelektrickém jevu, který spočívá ve spojení dvou vodičů z různého kovu, které mají různou teplotu, a prochází jimi elektrický proud. Při rozpojení obvodu, jsme schopni změřit termonapětí v řádu milivoltů, které je dáno teplotním rozdílem mezi spoji. [1]

Termonapětí je dáno kvadratickou funkcí teploty, což lze vyjádřit pomocí vztahu:

$$U_{AB} = k \cdot (t_a - t_b) \quad (1.2)$$

U_{AB} - termonapětí mezi referenčním a měřicím bodem termočlánu,

t_A a t_B jsou teploty těchto bodů;

k - kalibrační konstanta (podle typu termočlánu)

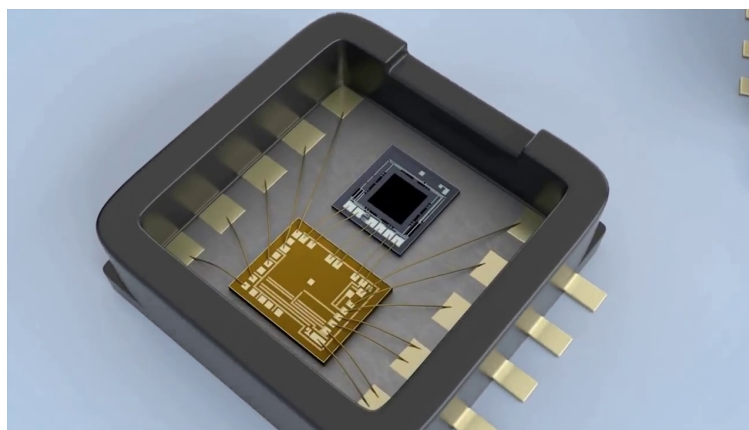
Nejpoužívanější materiály: měď-konstantan, mangan-konstantan

Přesnost měření teploty pomocí termočlánu závisí na citlivosti voltmetru měřícího termonapětí a na přesnosti měření teploty studeného konce termočlánu. [2]

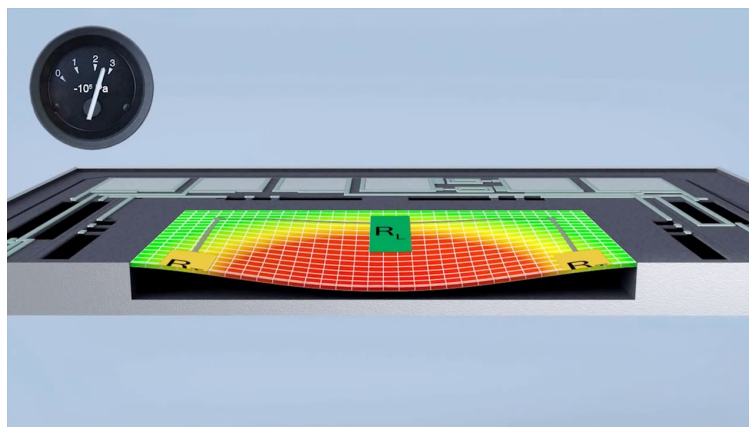
1.2 Měření atmosférického tlaku

Atmosférický tlak je síla, kterou působí atmosféra planety na jednotkovou plochu v daném místě. Atmosférický tlak dosahuje nejvyšších hodnot při hladině moře a s rostoucí výškou klesá. Atmosférický tlak není stálý, ale kolísá na daném místě zemského povrchu kolem určité hodnoty. [4]

V elektrických senzorech atmosférického tlaku se nejčastěji využívá Piezo-Rezistivního efektu. To znamená, že senzor obsahuje membránu z piezo-rezistivního materiálu. Vlivem atmosférického tlaku se membrána deformuje a tím mění svůj odpor. Proměnný odpor je zapojen do měřicího můstku, kdy malý vliv změny odporu, má velký vliv na můstkovém napětí. Toto napětí je dále zpracováno A/D převodníkem.



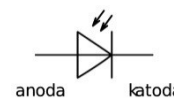
Obr. 1 3D model tlakoměru od výrobce Bosch



Obr. 2 3D model piézo membrány – deformace materiálu vlivem tlaku

1.3 Měření intenzity osvětlení

Nejčastěji se v senzorech intenzity světelného záření používají Fotodiody. Fotodiody je v elektrotechnice součástka, která reaguje na osvětlení. Princip fotodiody je založen na hradlovém fotoefektu. Do oblasti přechodu PN proniká elektromagnetické záření, které generuje páry elektron - díra.



Osvětlený přechod PN diody je vodivý i v závěrném směru a sám se stává zdrojem napětí. Tohoto jevu se využívá k přímé přeměně energie světelného záření na energii elektrickou. [5]

Při neosvětleném stavu je VA charakteristika fotodiody stejná jako u běžné diody. Vliv osvětlení PN přechodu je zřejmý ze záporné části charakteristiky, kdy dochází ke zvyšování anodového proudu v závislosti na osvětlení.

Fotodiody je dále vhodným způsobem připojena k A/D převodníku. Data z převodníku jsou dále zpracovávána a předána přes komunikační rozhraní

1.4 Měření srážek

Jeden z používaných principů měření srážek spočívá k naplňování váhy, která má přesně definovaný objem kapaliny, při které se překlápí. Po překlacení se kapalina vylije a je generován impulz. Princip generování impulzu spočívá v umístění jazýčkového relé do pevné části zařízení. V pohyblivé části váhy se nachází magnet, který při překlápění mívá jazýčkové relé.

Hodnota srážek při naplnění misky těsně před překlacením činí 0.2794 mm na 1m³



Obr. 3 Srážkoměr ze setu WH1080 + Vnitřní schéma

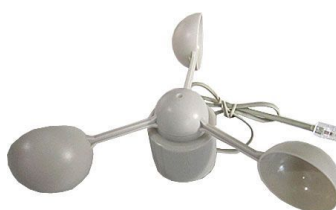
Typů zařízení pro měření srážek existuje mnoho další typů. Zde byl uveden konkrétní případ srážkoměru využitý v této práci

1.5 Měření rychlosti a směru větru

Pro měření rychlosti větru se často používá zařízení zvané Anemometr (z řeckého anemos = vítr).

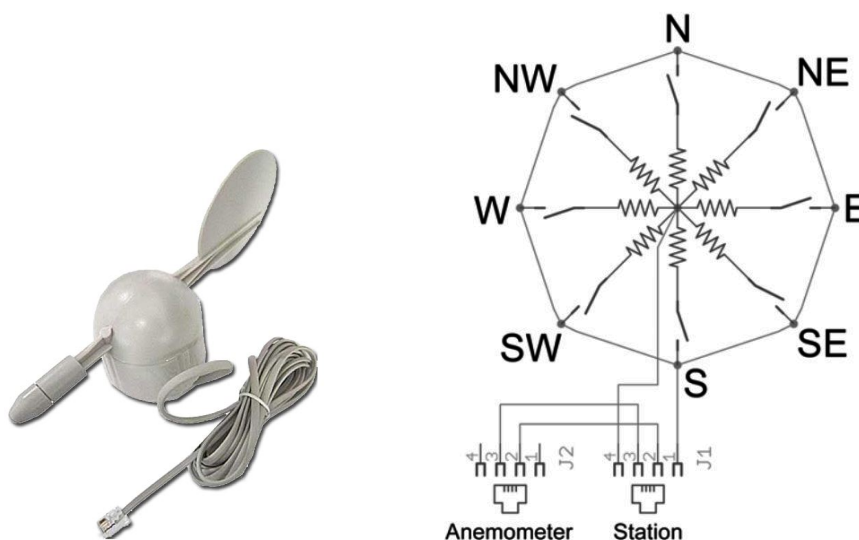
Mechanický Anemometr - Energie větru se přenáší na konstrukci, kterou vítr otáčí, rotuje či vychyluje z ustálené polohy.

Vnitřní konstrukce se skládá ze statoru, kde je umístěno jazýčkové relé s připojenými vodiči. A z rotoru, na kterém je připevněn magnet. Při každém otočení o 180° se magnet přiblíží jazýčkovému relé, jež je vzápětí na krátkou dobu sepnuto. Výstup je tedy impulzní, a výslednou rychlost otáčení udává frekvence impulzů.



Obr. 4 Anemometr ze setu WH1080

Využitý detektor směru větru aneb větrná směřovka je založena podobným principu. Také jsou zde použity magnety na rotoru a jazýčková relátka na statoru. Jak se směřovka otáčí, dojde k sepnutí jednoho kontaktu a tím se změní odpor senzoru. Vnitřní schéma směřovky vypadá následovně na obr. 5. Po doplnění pull-up odporu nám vzniká dělič napětí, jehož výstupní napětí závisí na aktuální poloze směřovky.



Obr. 5 Detektor směru ze setu WH1080 + Vnitřní schéma

2 Popis konkrétních senzorů

2.1 Teploměr SHT31

Senzor SHT31 od výrobce Sensirion je kombinovaný senzor teploměr/vlhkoměr. Je velmi populární pro jeho poměr cena/výkon. Jeho největší předností je přesnost měření teploty s odchylkou $\pm 0,2\text{ }^{\circ}\text{C}$ a měření relativní vlhkosti s odchylkou $\pm 2\text{ }\%$. Výrobce nabízí řadu SHT3x, ve které se vyskytují modely: SHT30, SHT31, SHT35. Čím vyšší číslo, tím lepší parametry. První dva jsou cenově skoro totožné, proto se vyplatí zakoupit o něco lepší SHT31, který má z této řady nejlepší poměr cena/výkon. Model SHT35 je opravdu tím nejlepším z řady, ale jeho cena je vyšší a také horší dostupnost na trhu. Jeho přesnost dosahuje $\pm 0,1\text{ }^{\circ}\text{C}$, RH $\pm 1,5\text{ }\%$.

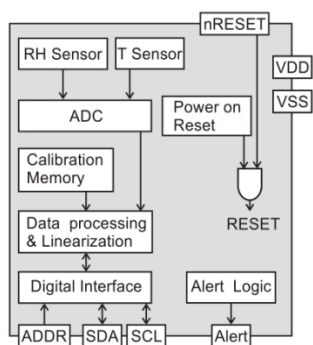


Obr. 6 čip SHT31

Tab. 1 Základní parametry SHT31 [6]

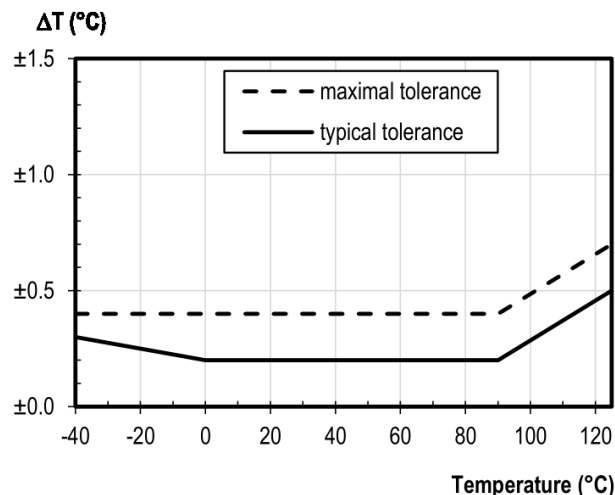
Napájecí napětí	2,4 – 5,5 V (3,3 V doporučeno)
Komunikační rozhraní	I2C (1MHz, 2 volitelné adresy)
Typická přesnost	$\pm 0,2\text{ }^{\circ}\text{C} \pm 2\text{ }\%$ RH
Technologie výroby	CMOS (CMOSens)
Další vlastnosti	Automatická kalibrace/linearizace, teplotní kompenzace – již z výroby
Rozlišení	0,015 $^{\circ}\text{C}$; 0,01 $\%$ RH
Volba I2C adresy	0x44 – ADR pin – LOW 0x45 – ADR pin – HIGH
Proudový odběr	Nečinnost - 45 μA , Měření - 1,5mA
Cena (3/2018)	4 USD

I když originální datasheet senzoru uvádí široký rozsah napájecího napětí, z osobní zkušenosti doporučuji ho napájet napětím 3,3V a stejně tak logické úrovně 3V3. Při napájecím napětí 5V a s logikou TTL 5V se obvod choval nestabilně, neboli v určitých mezích teplot přestával pracovat.



Z blokového schématu senzoru můžeme vidět, že na čipu integrovaného obvodu se nachází dva analogové senzory, podle změřených veličin mění svůj odpor, který je převeden na napětí a dále zpracován A/D převodníkem. Data z převodníku jsou zpracovány a upraveny (linearizace), také v závislosti na použitém napájecím napětí. Dále jsou data předány komunikačnímu rozhraní I2C, které data posílá na společnou sběrnici. Senzor má taky výstupní pin zvaný „Alert“, který slouží k oznámení speciální události.

Obr. 7 Blok. Schéma SHT31



Obr. 8 Graf přesnosti změřené veličiny v závislosti na teplotě – SHT31

Z grafu přesnosti v závislosti na teplotě můžeme vidět, že senzor podává nejlepší výsledky v rozmezí 5°C až 60°C. Z praktického hlediska se tedy při teplotách pod nulou a vysokých vlhkostech se přesnost měření zhorší, a pokud bude senzor vystaven těmto extrémním podmínkám delší dobu (např. > 80% RH po 60h), přesnost měření teploty a relativní vlhkosti se nadále bude zhoršovat.

2.2 Teploměr HTU21

Teploměr HTU21D(F) je levnější variantou teploměru SHT21, které jsou parametrově téměř shodné a pinoutem totožné. S dobrou přesností uvedenou v tabulce níže je jeho největší předností cena, oproti výše popisovanému senzoru SHT31, u které ho cena činí až 5 USD, tento senzor HTU21 stojí 3 USD. Proto byl zvolen jako sekundární senzor v projektu, který může nahradit primární senzor v případě výpadku nebo poruchy.



Obr. 9 3D model čipu HTU21

Podle měření které je uvedeno v poslední kapitole, je jeho přesnost skutečně velmi dobrá, vzhledem k jeho ceně, nicméně jeho výstup je posunut o určitou konstantu oproti skutečnosti. Proto je tato konstanta vykompenzována programově:

```
t_htu = t_htu_raw+0.15;
```

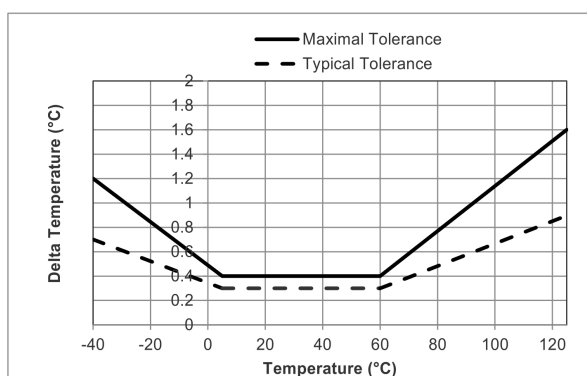
Z úryvku kódu programu vyplývá, že k výsledku je přičtena konstanta 0,15 °C. Jako referenční měřidlo byl použit primární senzor projektu SHT31, který má parametrovou přesnost $\pm 0,2^\circ\text{C}$

Z osobní zkušenosti při dlouhodobém testování mužů konstatovat, že tento senzor se choval nejstabilněji ze všech testovaných senzorů. V extrémních podmínkách velké relativní vlhkosti (- 18°C, 98% RH) dražší senzor SHT31 občas podával nesprávný výstup dat (NAN – not a number). A paradoxně tento levnější senzor nikoliv.

I z tohoto důvodu byl zvolen zástupný model senzoru, kdy pokud jeden senzor selže, zastoupí jej další. A protože jsou v projektu obsaženy celkem 3 teploměry (včetně tlakoměrů BMP280, který měří i teplotu). Až když všechny tři senzory selžou, což je sice nepravděpodobné ale možné, bude muset nastat nějaká vnější událost. Například předání informace kontroléru a odeslání automatického varovného emailu.

Tab. 2 Základní parametry HTU21 [7]

Napájecí napětí	1,5 – 3,6 V (3,3 V typ.)
Komunikační rozhraní	I2C (1 pevná adresa)
Typická přesnost	$\pm 0,3^\circ\text{C} \pm 2\% \text{ RH}$
Další vlastnosti	Kalibrace, linearizace, teplotní kompenzace – již z výroby
Rozlišení	0,04 °C; 0,04 % RH
I2C adresa	0x40
Proudový odběr	Sleep - 1μA, Měření - 450μA
Cena (3/2018)	3 USD



Obr. 10 Graf přesnosti změřené veličiny v závislosti na teplotě – HTU21

2.3 Tlakoměr BMP280

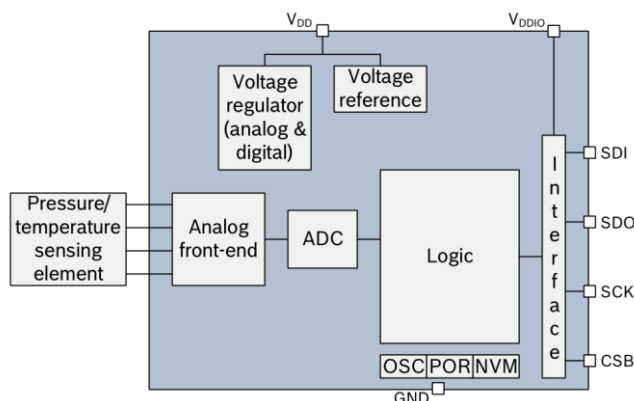
Senzor BMP280 od firmy Bosch je kombinovaný tlakoměr s teploměrem. Jeho princip funkce je založen na piézo-rezistivní membráně. Jeho rozměrem a spotřebou je vhodný zejména do mobilních zařízení. Teploměr se používá v obvodu především ke korekci výpočtu atmosférického tlaku. Nicméně lze jej použít k měření teploty s dobrou přesností.



Obr. 11 Podoba čipu BMP280

Tab. 3 Základní parametry BMP280 [8]

Napájecí napětí	1,7 – 3,6 V (3,3 V typ.)
Komunikační rozhraní	I2C, SPI
Absolutní přesnost (0-40°C)	± 1 hPa
Relativní přesnost (25°C)	± 0.12 hPa
Rozsah měření	300 – 1100 hPa
Absolutní přesnost teploty	± 1,0 °C
Max. frekvence měření	157 Hz
Rozlišení	0,16 Pa; 0,01°C
I2C adresa	0x76
Proudový odběr	2.7μA (fr=1Hz)
Cena (3/2018)	1 USD



Obr. 12 Blokové schéma senzoru BMP280

Senzor BMP280 se skládá z piézo-odporového snímacího elementu tlaku. Dále potom A/D převodníku zajišťující převod dat a kompenzaci hodnot. Dále obsahuje kombinované rozhraní I2C / SPI.

Senzor má 3 pracovní režimy:

- Sleep mode (Spánek),
- normal mode
- forced mode.

V režimu spánku přejde obvod do nečinnosti a jeho spotřeba se sníží na naprosté minimum $0.1\mu\text{A}$, což umožní bezproblémový provoz i na bateriové aplikaci. Normální režim zahrnuje automatizovaný režim trvalé cyklování mezi měřením a neaktivní pohotovostní dobou. V nuceném režimu (Forced mode) je provedeno jedno měření. Jakmile měření skončí, snímač se vrátí zpět do režimu spánku.

Podobně jako v případě senzoru HTU21, zde dochází k určitému posuvu výsledné hodnoty, proto je nutné uvést v kódu programu následující korekci. Referenční hodnota je opět vztažena k teploměru SHT31.

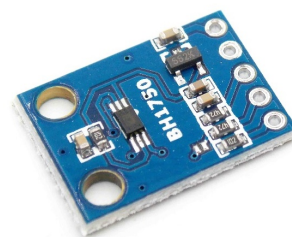
```
t_bmp = t_bmp_raw-1.73;
```

2.4 Luxmetr BH1750

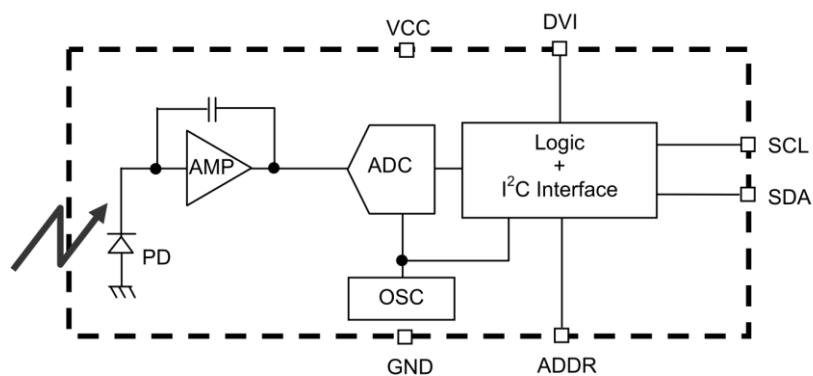
Senzor intenzity osvětlení (luxmetr) BH1750FVI od výrobce ROHM Semiconductor pro sběrnici I2C. Jeho využití najdeme především pro měření intenzity osvětlení a následné zpracování dat, např. regulace jasu LCD nebo podsvit kláves mobilních zařízení. Jeho výhodou je široký měřicí rozsah ($1 - 65535 \text{ lx}$).

Tab. 4 Klíčové vlastnosti senzoru BH1750 [9]

Napájecí napětí	1,7 – 4,5 V (3,3 V typ.)
Komunikační rozhraní	I2C
Rozsah měření	1 – 65535 lx
Max. perioda měření	120 ms
Rozlišení	1 lx
Volba I2C adresy	0x23 – ADR pin – LOW 0x5C – ADR pin – HIGH
Proudový odběr	$120\mu\text{A}$ (při $E=100\text{lx}$)
Cena (3/2018)	1 USD



Obr. 13 Modul BH1750



Obr. 14 Blokové schéma senzoru BH1750

Blokové schéma se skládá z foto-diody, která byla zvolena s ohledem na citlivost lidského oka na světlo. Dále je zde operační zesilovač, který zajistí převod proudu fotodiodou na napětí. Toto napětí je dále zpracováno 16-bit A/D převodníkem. Oscilátor pro interní logiku disponuje frekvencí 320 kHz.

3 Komunita „MySensors“

„MySensors“ je opensource hardwarová a softwarová komunita, která se zaměřuje na DIY – do it yourself - domácí automatizaci a internet věcí („Internet of things“). Komunita pomáhá svým členům vyrábět a oživovat cenově dostupné senzory připojené k mikrokontrolérům typu Arduino, ESP8266, Raspberry PI společně s bezdrátovými rádiovými moduly NRF24 nebo RFM69 atp. Celý web Mysensors.org je napsán na srozumitelné úrovni, což ocení zejména naprostí začátečníci v oboru, např. články „Getting started“, „Conect the radio“. Web obsahuje mnoho příkladů k různým druhům senzorů. Napsané knihovny komunitou „MySensors“ umožňují uživateli jednoduché programování MCU a připojených senzorů. Např. jedním příkazem „send“ provedeme posílání hodnoty senzoru do kontrolérů. Funkce v knihovnách už se postarají pro správný formát dat pro rádiový modul a také ověření, že přenos proběhl v pořádku. Nebo dalším příkazem „request“ si vyžádáme hodnotu z kontroléru a můžeme s ním dále pracovat. Ale o tom bude popsáno níže, zde je pouze demonstrováno jednoduchost programování při použití těchto knihoven. [2]



Obr. 15 Logo komunity MySensors

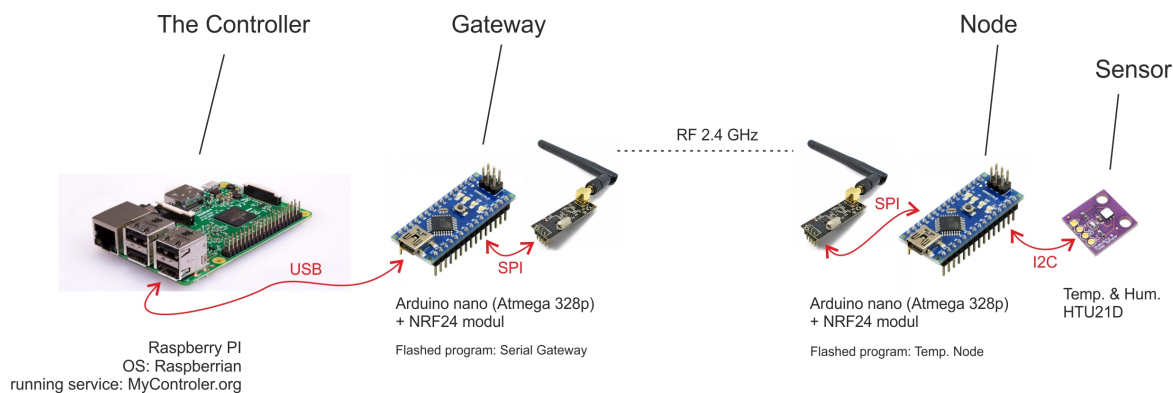
3.1 Topologie sítě

V této části bude vysvětlena topologie sítě „MySensors“, a vysvětlen rozdíl mezi pojmy Sensor, Node a Gateway.

Sensor je v síti člen, který posílá hodnoty dat tzv. příkaz SET. V praxi tedy bývá zařízení, které nastavuje (posílá) jednu nebo více proměnných. Např. v případě teploměru bude posílat proměnné teploty (V_TEMP – variable temp) a proměnou vlhkosti (V_HUM – variable humidity). Označením sensor můžeme také myslet virtuální senzor, který fyzicky neexistuje, ale je vytvořen programově. Např. Uptime data, průměrné hodnoty z více senzorů atd.

Všechny sensory musí být připojeny do nadřazené stromové struktury, což v praxi bývá nejčastěji tzv. node – uzel. Toto zařízení, nejčastěji jednoduchý 8bit mikro-kontrolér sbírá data ze senzoru, odesílá je dále do vyšších míst stromové struktury. Systém odesílání závisí na druhu zvoleného rozhraní pro přenos dat. V době populárních bezdrátových technologií, se nejčastěji setkáváme v síti „MySensors“ s rádiovým modulem NRF24 – pracující na frekvenci 2,4 GHz.

Dalším nadřazeným členem v síti je tzv. Gateway (brána). Tento člen zajišťuje převod dat z jednoho komunikačního rozhraní do druhého. Opět příkladem z praxe je převod dat z rádiového modulu, který přijmul data z dostupných uzlů (senzorů). Do rozhraní sériové linky (UART) společně s UART/USB převodníkem, což zajistí snadné připojení k serveru (Controleru), kde bude probíhat záznam dat.



Obr. 16 Nejjednodušší příklad sítě „MySensors“

Na obr. 16 je uveden příklad pro vysvětlení přenosu v síti „MySensors“,

Síť obsahuje následující členy:

- Sensor,
- node (uzel),
- gateway (brána),
- controler (server).

Jak již bylo řečeno Node-Uzel (dále jen jako uzel) je člen v síti, ke kterému jsou připojeny senzory a komunikační rozhraní. V tomto případě byl vybrán radiový modul nRF24 pro demonstraci. Jako senzor byl vybrán Teploměr HTU21.

Po zapnutí všech členů sítě proběhne základní inicializace Gateway-Brány, která nashodí jako první a čeká na připojení dostupných uzlů. V tu samou dobu uzel posílá periodicky spárovací data, dokud brána na tento požadavek neodpoví. Po inicializaci a spárování Gateway-Node může začít přenos dat.

Příkaz SET/SEND, který je obsažen v knihovnách „MySensors“, voláme ve zdrojovém kódu uzlu. Zajišťuje přenos hodnoty veličiny z uzlu na bránu. Brána se dále stará pouze o předání zprávy z radiového rozhraní do UART rozhraní (USB). V kontroléru je nastavený port COM, na kterém se má naslouchat komunikaci. Tím pádem do kontroléru přijde zpráva ve stejném formátu, jako byla odeslána z uzlu.

3.2 API - Application Programming Interface

API (zkratka pro Application Programming Interface) označuje v informatice rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí, tříd či protokolů nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může programátor využívat. V případě „MySensors“ rozdělujeme terminologii na Library API a Serial API. Library API obsahuje veškeré knihovní funkce, které můžeme v programu využívat. Tyto funkce a strukturu jednoduchého programu si ukážeme na praktickém příkladu.

Příklad pro snímání stavu dveří – Otevřené / Zavřené

```
#define MY_RADIO_NRF24
#include <MySensors.h>

#define OPEN 1
#define CLOSE 0
#define CHILD_ID 1

MyMessage msg(CHILD_ID, V_TRIPPED);
uint8_t value = OPEN;

void presentation()
{
  present(CHILD_ID, S_DOOR);
}

void loop()
{
  value = value == OPEN ? CLOSE : OPEN;
  send(msg.set(value));
  sleep(10000);
}
```

V hlavičce kódu musí být připojena knihovna MySensors.h, dále musíme zadefinovat rozhraní, po kterém bude uzel (node) komunikovat s bránou (GW). To je zajištěno definicí #define MY_RADIO_NRF24, neboli přenos modulem NRF24. Dále je zadefinované tzv. Child ID, neboli ID dceřiného sensoru, v tomto příkladu máme pouze jeden senzor, takže postačí jednoduché označení. V případě více různých senzorů musíme zvolit rozlišitelný název pro každý senzor. Dále musíme vytvořit zprávu a vyplnit její strukturu, tj. Child-ID – 1, a typ proměnné V_TRIPPED – Variable tripped, neboli proměnná která bude nabývat dvou stavů, zapnuto/vypnuto respektive otevřené/zavřené dveře.

Každý program musí mít funkci presentation(), ve které se po zahájení programu před smyčkou loop(), provede odeslání základních informací automaticky: node-id, node-name. A také manuálně, v našem případě Child-ID a S_DOOR, tzn., že se bude jednat o dveřní senzor. A toto označení je součástí knihoven a proto se při každém startu uzlu pošlou data do kontrolérů, a ten proměnné automaticky zaregistruje jako dveřní senzor.

V hlavní smyčce `loop()` pak již probíhá jen periodické zjišťování stavu dveří a následné odeslání zprávy příkazem „send“, který má atributu „set“, což znamená, že předchozí hodnota bude přepsána novou.

O definici zprávy, kterou budeme posílat z uzlu (node) do controlleru, tzv. `MyMessage` zahrnujeme do sekce Serial API. Tato zpráva, která má následující formu při posílání rozhraním pomocí Strings:

`node-id ; child-sensor-id ; command ; ack ; type ; payload \n`

Obr. 17 Struktura posílané zprávy (`MyMessage`)

- Node-id – Identifikační číslo uzlu,
- child-sensor-id – Identifikační číslo senzoru, které je dceřiným zařízením daného uzlu,
- command – příkaz co se má s daty provést,
- ack – Žádost o poskytnutí zpětné vazby, zda přenos proběhl v pořádku,
- type – Úzce souvisí s položkou command, více popsáno na webu,
- payload – Hodnota ze senzoru, která se ve zprávě přenáší.

Message Part	Comment
<code>node-id</code>	The unique id of the node that sends or should receive the message (address)
<code>child-sensor-id</code>	Each node can have several sensors attached. This is the child-sensor-id that uniquely identifies one attached sensor
<code>command</code>	Type of command sent - See table below
<code>ack</code>	The ack parameter has the following meaning: Outgoing: 0 = unacknowledged message, 1 = request ack from destination node, Incoming: 0 = normal message, 1 = this is an ack message
<code>type</code>	Depending on command, the type field has different meaning. See tables below
<code>payload</code>	The payload holds the message coming in from sensors or instruction going out to actuators.

Obr. 18 Originální popis Serial API, z webu MySensors.org

Type	Value	Comment
presentation	0	Sent by a node when they present attached sensors. This is usually done in the <code>presentation()</code> function which runs at startup.
set	1	This message is sent from or to a sensor when a sensor value should be updated
req	2	Requests a variable value (usually from an actuator destined for controller).
internal	3	This is a special internal message. See table below for the details
stream	4	Used for OTA firmware updates

Obr. 19 Originální popis příkazů (`command`), z webu MySensors.org

3.3 Komunikační rozhraní

3.3.1 Bezdrátový modul nRF24L01+

Tento modul je v komunitě „MySensors“ velmi oblíbený a nejpoužívanější. Jedná se o bezdrátový modul, který své data posílá pomocí svého speciálního protokolu na frekvenci 2,4 GHz. I když pracuje ve stejném pásmu jako „Wi-Fi“ nebo „Bluetooth“, jeho specifickou modulací a nižší přenosovou rychlostí dosahuje poměrně vysoký dosah v otevřeném prostředí. Jeho dosah samozřejmě rapidně klesá se vyšším počtem „cihlových zdí“ a rušení z okolních zařízení pracujících na stejné frekvenci.

Modul existuje ve více variantách provedení, v této práci byl otestován modul v provedení s externí anténou RSMA. Existuje také provedení s integrovanou anténou, které vyhoví na kratší vzdálenosti a společně s kompaktními senzory.



Obr. 20 nRF24

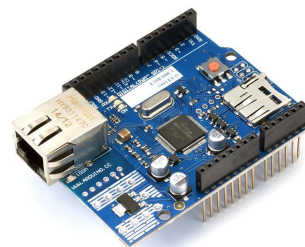
Tab. 5 Základní parametry modulu nRF24L01+

Napájecí napětí	3,3 V
Komunikační rozhraní	SPI
Radiová frekvence	2,4 GHz, ISM pásmo
Přenosové rychlosti	250 kbps, 1Mbps, 2Mbps
Proudový odběr - komunikace	14mA (při P = 0dBm)
Proudový odběr – pohotovostní režim	26 μ A
Proudový odběr – kom.,max.P	150mA (při P = 20dBm)
Cena (3/2018)	2 USD

3.3.2 Ethernet shield s čipem W5100

Jedná se o ethernetový modul pro desky „Arduino UNO“, jejich pinout je kompatibilní a stačí jej pouze zasunout. Je to nejjednodušší způsob, jak připojit desky „Arduino UNO“ respektive jednoduché 8bit MCU Atmel do internetu. Toto seznámení s modulem zde uvádím proto, že bude využito, jako jedno z možností komunikačního rozhraní na univerzální desce jednotky sběru dat.

Modul obsahuje čip W5100, který disponuje plnou podporou TCP/IP protokolu, tj. TCP, UDP, IPv4, PPPoE atd. Čip podporuje rychlosti ethernetové linky 10/100 Mbps. Propojení s procesorem „ATMega“ je zajištěno sběrnici SPI.



Obr. 21 Eth. modul

Tab. 6 Klíčové parametry čipu

Napájecí napětí	3,3 V
Komunikační rozhraní	SPI
Support Hardwired TCP/IP Protocols	TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
Podporované normy	10BaseT/100BaseTX Ethernet PHY embedded
	Support Auto Negotiation (Full-duplex and half duplex)
Cena (3/2018)	5 USD



Obr. 22 čip W5100

Obsluhu celého modulu provádíme pomocí vyvinutých knihoven „Ethernet.h“, ve kterých se nachází spousta příkladů s tímto modulem. Modul obsahuje také patičku microSD karty, která je připojena společnou SPI sběrnici s vývodem CS – Chip Select. Ta sice v projektu nebude využita, ale může se snít počítat do budoucna, s případným logováním dat.

Praktická nepříjemnost při práci s tímto modulem, je občasný nepovedený start při zapnutí napájení, proto byla na desce přidána funkce automatického resetu při zapojení napájení

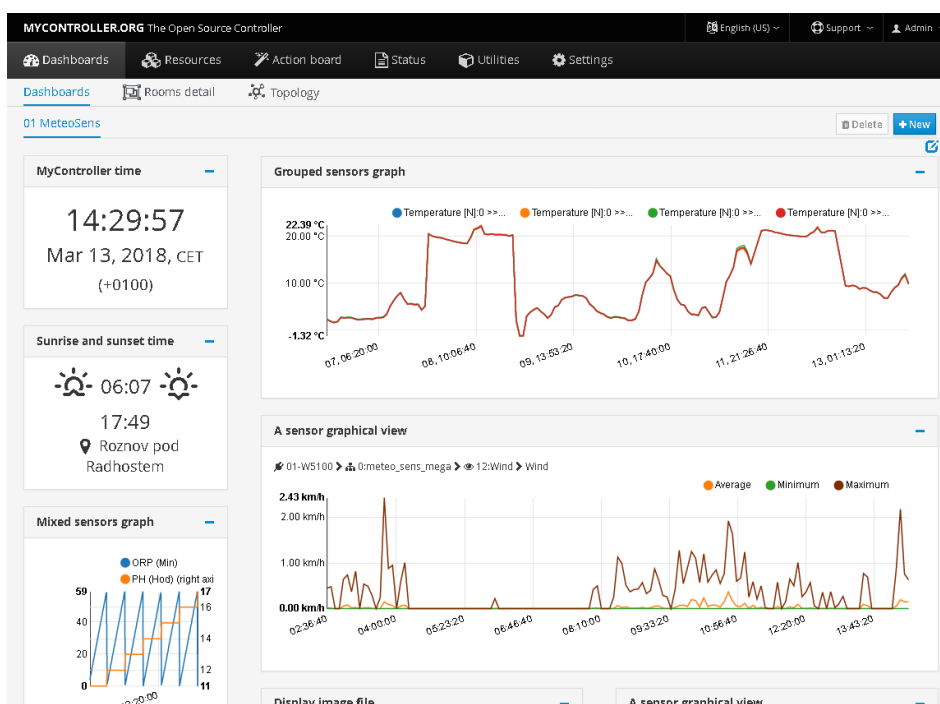
3.4 The Controller

Důležitá poznámka: Označení „Controller“ se může plést s označením „Mikrokontroler“. Proto tedy pro upřesnění:

- Controller (kontroler) – Označení serveru, na kterém se ukládají data ze sítě „MySensors“.
- Mikrokontroler (MCU) – Malý jednočipový počítač, fakticky každý uzel v síti „MySensors“

Jak již bylo zmíněno výše, „Controller“ je serverová služba, která ukládá data ze senzorů sítě „MySensors“, nebo je schopna řídit členy v síti, např. relé. Podle použitého kontroléru, se pak výsledky dají zobrazit v grafu, seznámech atd. Pro komunitu „MySensors“ bylo vyvinuto více kontrolérů od různých vývojových skupin. Ty se liší zejména v napsaném programovacím jazyku, vzhledem, chováním (stabilitou) a přidavnými speciálními funkcemi. Základní funkcionalita je však stejná, a tak výběr vhodného kontroléru je založen spíše na osobních preferencích. V rámci této diplomové práce byl zvolen kontrolér – MyController.org.

Jedná se o „opensource“ kontrolér vyvíjen nadšencem z Indie, je napsán v programovacím jazyku - Java. Proto je nezbytné mít pro jeho provoz nainstalován „Java runtime enviroment“. V balíčku s kontrolérem je i obsažen Apache webový server, tudíž jeho instalace je jednoduchá. Kompatibilní je s operačními systémy Windows, Linux respektive s Debian/Raspberrian pro provoz na vývojové desce Raspberry PI.



Obr. 23 Uživatelský vzhled aplikace MyController.org

4 Projekt: „MeteoSens“

Název projektu „MeteoSens“ byl zvolen hlavně z důvodu přehlednosti všech uzlů v kontroléru.

Jak již bylo řečeno v úvodu, pro ověření funkce a kvality senzorů bude postavena meteostanice, která bude měřit následující veličiny:

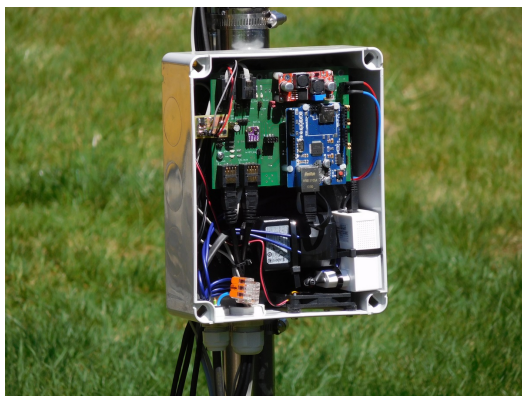
- Teplotu,
- relativní vlhkost,
- rosný bod,
- atmosférický tlak (absolutní, vztažený k hladině moře),
- intenzitu osvětlení ze slunce,
- rychlost a směr větru,
- srážky.

Senzory teploty, vlhkosti a tlaku budou umístěny v radiačním krytu, který bude mít aktivní odvětrávání pomocí ventilátoru. Veškeré díly meteostanice budou na společné stožárové konzoli, tj. radiační štít, anemometr, srážkoměr a jednotka s elektronikou ve venkovní vodotěsné krabici. Přenos z čidel bude probíhat nativně sběrnici I2C přenášené po UTP kabelu. Přenos z čidel větru a srážek bude probíhat po klasické telefonní dvoulince/čtyř-lince. Pro napájení ventilátoru bude použita dvoulinka CYA 2x0.75.

4.1 Konstrukční provedení

Uvnitř venkovní krabice vysílací jednotky meteostanice jsou následující komponenty:

- Hlavní DPS – univerzální pro různé komunikační rozhraní. (nRF24 / Ethernet),
- napájecí zdroj 5V; 2,5A,
- adaptér PowerLine – pro přenos dat po rozvodné síti 230V,
- ventilátor,
- „Raspberry PI“ – pouze pro potřebu vzdáleného programování a vývoj, později bude vyjmuto.

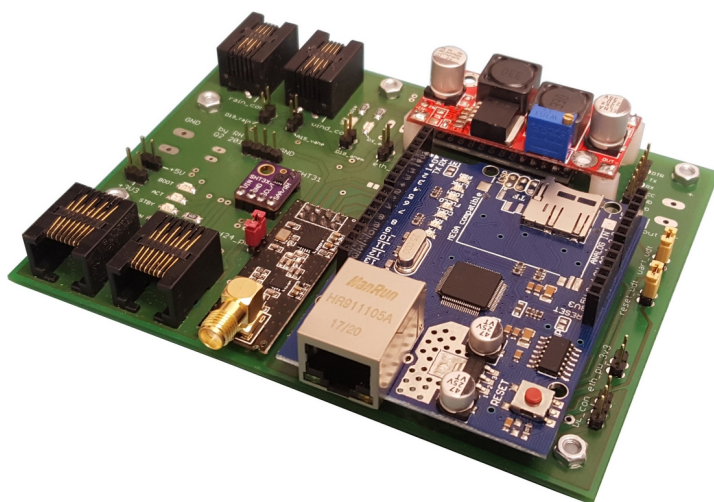


Obr. 24 Podoba vysílací jednotky meteostanice

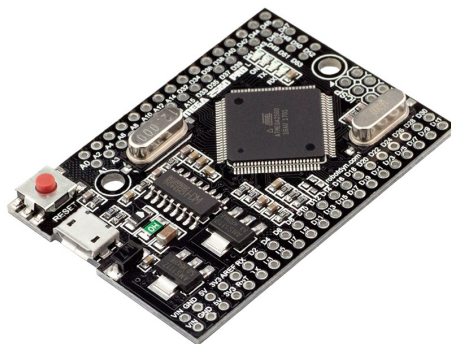
4.2 DPS jednotky sběru dat

Jednotka sběru dat obsahuje:

- Patici pro modul Mega Pro s procesorem ATmega 2560,
- patici pro Ethernetový modul s procesorem W5100,
- patici pro modul nRF24,
- patici pro obvod WatchDog respektive pinout desky Arduino pro mini,
- buck-boost říditelný měnič s driverem XL6009 pro napájení ventilátoru,
- indikační LED – STBY/ACT, BOOT, FAN,
- konektory RJ45 pro připojení I2C čidel,
- konektory RJ11 pro připojení analogových čidel.

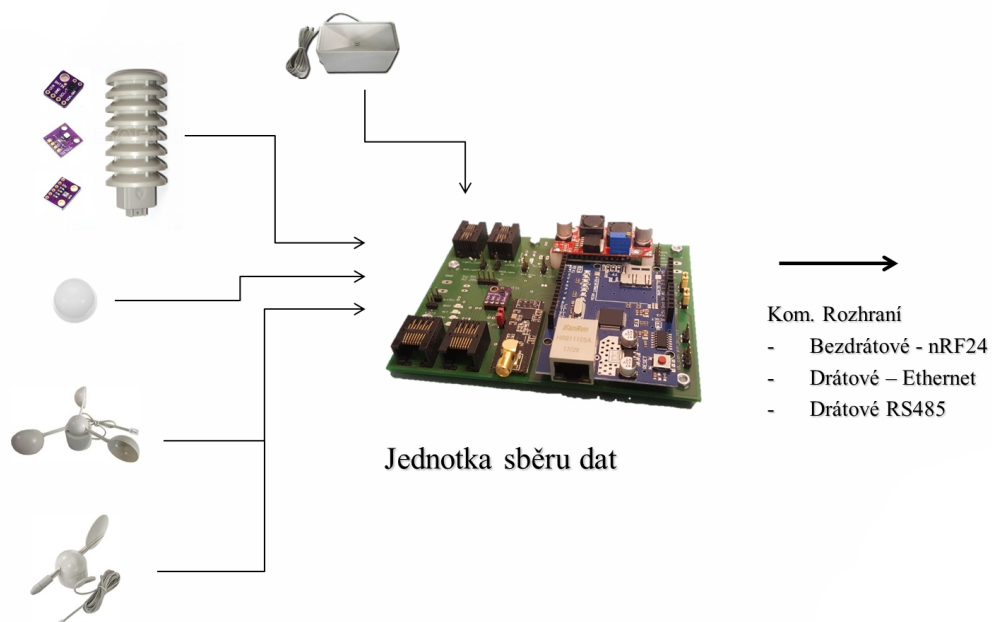


Obr. 25 Podoba jednotky sběru dat

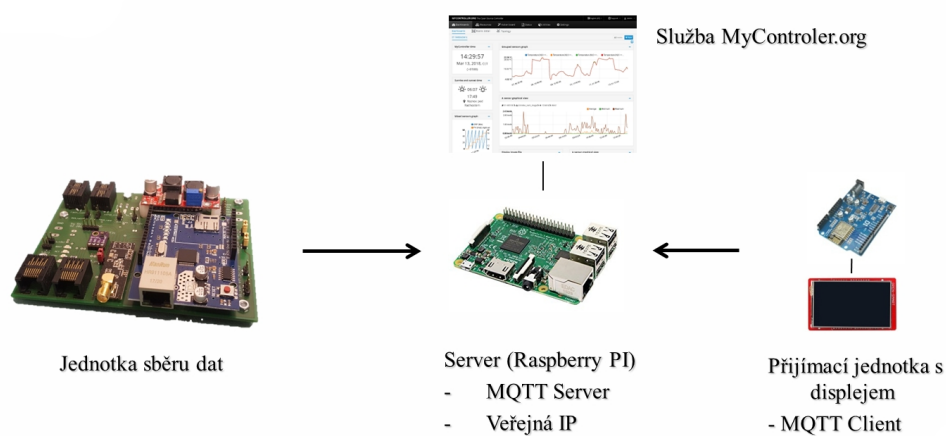


Obr. 26 Modul Mega-Pro-Mini s procesorem ATmega 2560

4.3 Pricipiální schéma systému



Obr. 27 Blokové schéma projektu „MeteoSens“



Obr. 28 Blokové schéma – připojení k serveru (kontroléru)

4.4 WatchDog

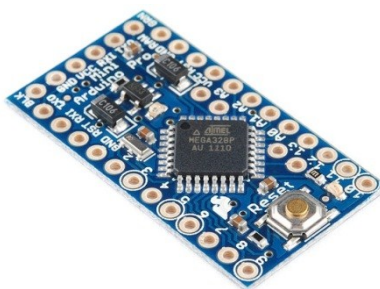
Watchdog nebo Watchdog timer, zkráceně WDT (z angličtiny – „hlídací pes“) je počítačová periférie, která resetuje systém při jeho zacyklení. K zacyklení systému může dojít v důsledku chyby v hardware nebo software systému.

Protože při stavbě a vývoji projektu byl kladen důraz na jeho stabilitu, byl implementován WatchDog do desky sběru dat. Procesory ATmega sice disponují hardwarovým WDT, nicméně jeho maximální nastavitelná doba, po které provede reset, činí 8s. Tato hodnota je příliš nízká, protože obnovení činnosti celého systému po přerušení napájení může trvat i více než 20s.

Po dlouhé úvaze byl zvolen externí hardwarový WatchDog. Při výběru vhodného řešení připadali v úvahu primitivní typy na bázi obvodu NE555, a speciálními jednoúčelovými obvody WDT. Nakonec bylo využito zcela jiné řešení, a to spočívá v naprogramování funkce watchdog do mikrokontrolérů ATmega 328p, který bude také součástí DPS jednotky sběru dat.

Jeho princip spočívá ve sledování aktivity na sériové lince procesoru ATmega 2560, kde probíhá výpis debugovacích informací. Pokud sériová linka nevykáže žádnou činnost po dobu 90s, provede se reset. Obvod také zajišťuje reset po připojení napájení.

Použití takového typu WatchDogu je sporné protože i ten může selhat, nicméně tato situace je mnohem nepravděpodobnější než v případě složitého programu projektu „MeteoSens“. Cena modulu s procesorem ATmega 328p činí 1USD, i to byl jeden z důvodů pro jeho výběr.



Obr. 29 Podoba modulu s procesorem ATmega 328p, známý také jako Arduino Pro Mini

Obvod WDT byl implementován v době kdy, se začínalo s procesorem ATmega 328p jako hlavním MCU jednotky sběru dat. Tento MCU disponuje operační pamětí 2kB, což při zaplnění přes 70% způsobovalo značnou nestabilitu. Např. tehdejší průměrný uptime dosahoval pouze 1-2 dny, poté se obvod zasekl a obvod WDT musel provést restart.

Protože nedostatek operační paměti i paměti dat komplikoval následný vývoj, byla zvolena varianta MCU ATmega 2560, který obsahuje 8 kB RAM a 256 kB paměti dat. Tento krok měl vliv i na stabilitu provozu a proto obvod WDT nemusí zasahovat tak často. V nenáročných projektech by mohl být zcela odstraněn. Nicméně byl zanechán i z důvodu jeho minimální ceny. Zatím tedy průměrný uptime nelze určit protože WDT ještě nemusel provádět reset. Aktuální maximální hodnota uptime činí 7 dní, poté byl proveden restart obsluhou nebo update programu.

4.5 Software

Díky knihovnám „MySensors“ odpadá nutnost programovat komunikační rozhraní, protože knihovny obsahují funkce, jak pro bezdrátový modul nRF24, tak i pro ethernetový modul W5100. Jejich provoz současně však není možný a jejich výběr provedeme za definováním:

```
#define MY_RADIO_NRF24
#define MY_GATEWAY_MQTT_CLIENT
```

Dále díky těmto knihovnám odpadá programování správy přenosu, funkce ACK – (acknowledgement) je již obsažena, a zaručuje zpětnou vazbu, pro vyhodnocení úspěšného přenosu zprávy.

Úkolem v rámci programování uzlu s procesorem ATmega 2560 bylo zajistit správné přečtení hodnoty ze senzoru a ošetřit všechny nežádoucí stavy. Následné také detekce nesmyslné hodnoty ze senzoru, např. pokud senzor změří správnou teplotu 20 °C, algoritmus mu nedovolí v dalším cyklu měření přečíst a poslat velmi odlišnou teplotu. Protože změna teploty neprobíhá skokově. Součástí programu je také omezení maximálních hodnot podle údajů teplotních rekordů v ČR.

Pokud by, tyhle procedury nebyly zajištěny, mohla by při výskytu chyby komunikace se senzorem být odeslána nesmyslná zpráva, např. venkovní teplota s hodnotou -127 °C

Ukázka základního ošetření přečtené hodnoty

```
float const horni_t_limit = 40.99; //rekord v CR 40.4
float const dolni_t_limit = -42.99; //rekord v CR - 42.2

void sht31_send(){

    t_sht_raw = sht31.readTemperature();

    if(t_sht_raw != 0){
        if(t_sht_raw < horni_t_limit && t_sht_raw > dolni_t_limit){
            t_sht = t_sht_raw;
            Serial.println("    SHT31:");
            send(sht_t_Msg.set(t_sht,2));
            delay(delay_time);
        }
    }
}
```

Z výše uvedené ukázky můžeme vidět hodnoty omezení teploty, toto omezení zajistí v případě překročení mezní hodnoty neposlání zprávy z uzlu do kontroléru. V případě spodní hranice je taková situace velmi nepravděpodobná. V případě horního omezení je situace pravděpodobná v případě přímého slunečního záření na radiální kryt, ve kterém se nepříznivě ohřeje vzduch. K eliminaci takového nežádoucího stavu, byl vybaven kryt ventilátorem. Ten zajistí nucenou cirkulaci vzduchu. Ověření takového řešení jako spolehlivé, však proběhne až v horkých letních dnech.

Dalším úkolem v rámci programování, bylo vytvořit sekvenci posílání hodnot, ve které mají přednost skokové veličiny před spojitými veličinami. Např. teplotu stačí odesílat každých 30s nebo i více. Ale rychlost větru se bude odesílat každou sekundu.

Následující ukázka programu představuje posílání hodnot přechtených z čidel ve smyčce a s prodlevou mezi jednotlivými veličinami. Jedná se o výňatek ze smyčky Loop(), a vymazání nedůležitých částí pro názornou ukázkou:

```
currentMillis = millis();
if (currentMillis - previousMillis_sec >= interval_sec) {
    previousMillis_sec = currentMillis;
    id++;
    Serial.print("id: ");Serial.print(id);
    Serial.println(" -----");

    if(id > 0){
        digitalWrite(41, HIGH);
        digitalWrite(38, LOW); //Stby LED
        wind_speed_send();
        wind_vane_send();
        uart_check();
    }

    if(id == -9){
        if(request_need == 1){
            if(rain_imp_cnt == 0){
                Serial.print("request:");
                request(RAIN_CHILD, V_VAR5);
            }
            request_need = 0;
        }
    }

    if(id == 2){
        sht31_send();
    }

    .
    .
    .

    if(id == 9){
        rain_send();
        id = 0;
    }
}
```

Princip prodlevy mezi odesíláním veličin je založen na funkci millis(). Zavoláním této funkce dostáváme hodnotu v milisekundách od spuštění MCU. Pokud je tato funkce volána ve smyčce loop() výsledná hodnota datového typu uint32 vždy odpovídá aktuálnímu času od spuštění MCU. Tím pádem tuhle funkci můžeme využít k vytvoření prodlevy. Pro zjednodušenou představu si zapíšeme aktuální čas do proměnné. Podmínku, kterou vytvoříme, bude platit pouze když:

```
if (currentMillis - previousMillis_sec >= interval_sec) {
    previousMillis_sec = currentMillis;
    id++;
}
```

Neboli program čeká, až uběhne doba od posledního provedení funkce. Dále pak jen inkrementujeme hodnotu ID, podle které se posílají jednotlivé skupiny veličin z čidel. [1]

4.5.1 Použité vývojové prostředí

Zvolený MCU je v podstatě menší varianta velké desky Arduino Mega, tzn., že použitý MCU je stejný, rozvržení desky je odlišné, nicméně je pinout kompatibilní. Použitý převodník USB/UART je odlišný, nicméně po instalaci příslušného ovladače, je plně podporován originálním prostředím Arduino IDE.

Prostředí Arduino IDE je vhodné zejména k programování desek Arduino, nicméně lze do něj doinstalovat podporu dalších vývojových desek. Např. Wi-Fi MCU s označením ESP8266 je po instalaci balíčku také podporován. Toto prostředí vyniká svou jednoduchostí, a je určené hlavně pro začátečníky. Postrádá pokročilé funkce vývoje softwaru, např. debug. Prostředí umožňuje si vytvářený software psát do záložek, které se do hlavního souboru (main.ino) includují automaticky.

Toto prostředí bylo nainstalované na malý počítač „Raspberry PI“, které bylo umístěné přímo ve venkovní krabici meteostanice. A tudíž vývoj softwaru a ladění probíhalo za chodu přes vzdálenou plochu (VNC)



Obr. 30 Vývojové prostředí Arduino IDE na OS: Raspberrian

4.5.2 Debuggovací log

Protože vývojové prostředí Arduino IDE neobsahuje plnohodnotný debug rozhraní. Jediný způsob pro ladění softwaru je sledování dat v sériové konzoli. Knihovny „MySensors“ umožňují funkci tzv. „Debugg log“. Jehož ukázka je zobrazena níže.

```
0;255;3;0;9;MCO:BGN:INIT GW,CP=R-NGA--,VER=2.1.1
IP: 192.168.3.43
0;255;3;0;9;MCO:BGN:STP

--- meteo_sens 1.1 ---
--- thingspeak version ---
--- build date: 17.4.2018 ---
--- by R.Habartik ---

0;255;3;0;9;MCO:REG:NOT NEEDED
0;255;3;0;9;MCO:BGN:INIT OK,TSP=NA
IP: 192.168.3.43
0;255;3;0;9;Attempting MQTT connection...

id: -11 -----
IP: 192.168.3.43
0;255;3;0;9;Attempting MQTT connection...
0;255;3;0;9;MQTT connected
0;255;3;0;9;Sending message on topic: mygateway1-out/0/255/0/0/17
0;255;3;0;9;Sending message on topic: mygateway1-out/0/255/3/0/11

id: -9 -----
request:0;255;3;0;9;Sending message on topic: mygateway1-out/0/13/2/0/28
0;255;3;0;9;Message arrived on topic: mygateway1-in/0/13/1/0/28
Received value from gw:4676
RAIN:
0;255;3;0;9;Sending message on topic: mygateway1-out/0/13/1/0/28

id: -7 -----
SHT31:
0;255;3;0;9;Sending message on topic: mygateway1-out/0/1/1/0/0
0;255;3;0;9;Sending message on topic: mygateway1-out/0/1/1/0/1

id: 1 -----
FAN_CONTROL:
0;255;3;0;9;Sending message on topic: mygateway1-out/0/19/1/0/37
0;255;3;0;9;Sending message on topic: mygateway1-out/0/19/1/0/51
fan_state_cover: 0
fan_state_case: 0
inboard_temp: 29.90
AVG: 16.72
LUX: 4511

id: 10 -----
W:
W_actual:0.91
0;255;3;0;9;Sending message on topic: mygateway1-out/0/12/1/0/8
```

Tato ukázka představuje typický start jednotky, který byl proveden úspěšně. Každý nový řádek, který je generován knihovnou, nikoliv funkcí „Serial.print“, začíná daty posílané zprávy podle formátu Serial API. Např. zpráva pro posílání dat ze senzoru na Server pomocí standartu MQTT:

```
0;255;3;0;9;Sending message on topic: mygateway1-out/0/255/0/0/17
```

Z logu je patrné že při zahájení programu proběhne inicializace v požadovaném nastavení, v tomto případě jako MQTT Gateway. Dále se získává IP z DHCP serveru, nebo je nastavena staticky. Po inicializaci sítě je proveden pokus o připojení MQTT serveru. Po úspěšném připojení se pošlou všechny data o připojeném uzlu: Název uzlu, verze knihoven atd. Jako další funkce, která musí proběhnout je tzv. „request“. Jedná se o načtení dat z kontroléru, která jsou zaznamenávaná formou přičítání. V konkrétním případě se jedná o množství spadlých srážek.

```
request:0;255;3;0;9;Sending message on topic: mygateway1-out/0/13/2/0/28
0;255;3;0;9;Message arrived on topic: mygateway1-in/0/13/1/0/28
Received value from gw:4676
```

Je to patrné z této části, kdy se uzel požádá o přenos dat – Sending message. A jako odpověď se vrátí zpráva s požadovanými daty. Z této zprávy je vybrána nesenná hodnota a je uložena do proměnné typu „Long“ (32bit).

Pokud procedura „request“ proběhne v pořádku, a zároveň hodnota nabývá ID>0, může započít periodické posílání dat ze senzoru do kontroléru podle daných ID. Frekvence změny hodnoty ID je nastavena 3s, ale lze ji programově libovolně nastavit.

Hodnoty teploty, vlhkosti a tlaku jsou tím pádem aktualizovány každých 10s, hodnoty rychlosti větru jsou posílány s každým novým ID, čili každé 3s. V případě hodnoty směru větru se posílá hodnota, která se nejčastěji vyskytovala v daném časovém rozmezí, konkrétně po dobu jedné minuty (Direction per min) a také po dobu jedné hodiny (Direction per hour).

4.6 Příjímací jednotka s displejem

Součástí projektu „MeteoSens“ bude také přijímací jednotka s displejem, jejím úkolem bude ukazovat všechny důležité veličiny v reálném čase. Data budou přenášena bezdrátově technologií Wi-Fi. Princip přenosu dat bude spočívat v nastavení MQTT serveru (brokeru) jako veřejný a dostupný z internetu. Na tento server se posílají veškeré data z meteostanice v tzv. topicu „mysensors-1“ tyto data budou pomocí skriptu předány (funkce „forward payload“) do topicu „mysensors-2“, na který bude připojena přijímací jednotka jako MQTT klient. Výhoda funkce forward payload je v řízení přeposílání dat, nastavením v kontroléru můžeme změnit přenášené veličiny nebo proces zcela zakázat. Protože je MQTT server veřejný může být přijímací jednotka umístěna naprosto kdekoliv, jen musí být zajištěno připojení do internetu. Náročnost na datový přenos je minimální.

Fyzicky se bude jednotka skládat z desky Wemos D1 s čipem ESP8266, který zajišťuje připojení do internetu. Dále budou data přeneseny po sériové lince na desku Arduino UNO s čipem Atmega328p. Dále jsou data zobrazeny na dotykovém barevném TFT displeji o rozlišení 420x240px. Čistší řešení by spočívalo v přímém připojení displeje na čip ESP8266, který potřebné SPI rozhraní má. Nicméně pro daný displej neexistují knihovny, a vytvoření nových knihoven je mimo rozsah této diplomové práce. Ale protože obě desky jsou pinout kompatibilní, lze je po drobné úpravě přímo zasunout do sebe.

Nyní je zapotřebí vymyslet přenosový protokol mezi deskami. Z časové nedostupnosti byl vytvořen ten nejprimitivnější, jak to jen jde. Data se posílají přímo ve formě „strings“ spolu se speciálním znakem z ASCII tabulky. Tento speciální symbol určuje ID dané přenášené veličiny. Na přijímací straně desky Arduino UNO je softwarový dekodér, který podle speciálního symbolu určí ID, a číselné data uloží do příslušné proměnné „strings“. Speciální znaky byly zvoleny z důvodu, že na sériové lince se vyskytují také debuggovací data čipu ESP8266, a tyto data nesmí narušit přenos dat z jedné desky do druhé.

```
no wifi found, reconnect after 1s
reconnect
....scandone
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
aid 1
cnt

connected wifi, channel 6
dhcp client start...
...ip:192.168.43.150,mask:255.255.255.0,gw:192.168.43.1
.IP: 192.168.43.150
IP: 192.168.43.150

??80.60@
**18.30@
##1015.20@
!!4.20@
```

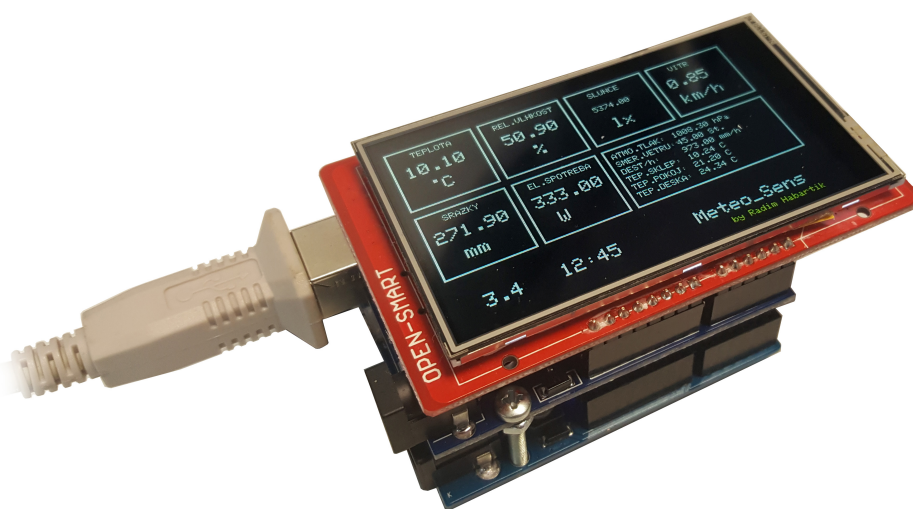
UART Log. ESP8266

Následující tabulka ukazuje volbu speciálních symbolů a jejich HEX podobu v ASCII tabulce.

Tab. 7 Přiřazení ID symbolu z ASCII tabulky

id	Variable type	fwd sensor-id	fwd-name	sensor-description	display-mcu-id	transfer-symbol	sym-hex
1	V_TEMP	1	fwd1	temp-avg-main	1	!	21
2	V_HUM	2	fwd2	sht31-hum	2	?	3F
3	V_PRESSURE	3	fwd3	bmp280	3	#	23
4	V_LEVEL	4	fwd4	Light BH1750	4	\$	24
5	V_WIND	5	fwd5	wind_actual	5	%	25
6	(V_GUST)V_Direction	6	fwd6	Direction per min	6	&	26
7	V_RAIN	7	fwd7	Rain_total	7	"	27
8	V_RAIN_RATE	8	fwd8	Rain per day	8	[5B
9	V_WATT	9	fwd9	Wattmetr actual	9]	5D
10	V_KWH	10	fwd10	Kwh	10	*	2A
11	V_VAR1	11	fwd11	temp-inboard	11	+	2B
12	V_VAR2	11	fwd11	temp-avg-2(terasa)	12	`	60
13	V_VAR3	11	fwd11	temp-avg-3(puda)	13	/	2F
14	V_VAR4	11	fwd11	temp-avg-4(sklep)	14	{	7B
15	V_VAR5	11	fwd11	temp-radim-pokoj	15	}	7D

Komunikace je jednosměrná, tedy bez žádné zpětné vazby ani kontrolního součtu, proto nedosahuje 100% spolehlivosti. Nicméně data se odesílají neustále periodicky, perioda odeslání jednoho typu veličiny činí cca 10s. Pokud tedy nastane chybný přenos, veličina se za 10s obnoví. Testováním se ukázalo, že i takto jednoduchý přenosový protokol je aktuálně funkční, a pokud v budoucnu nastanou problémy, bude potřeba přidat zpětnou vazbu, např. funkci ACK, kontrolní součet atp.



Obr. 31 Podoba přijímací jednotky

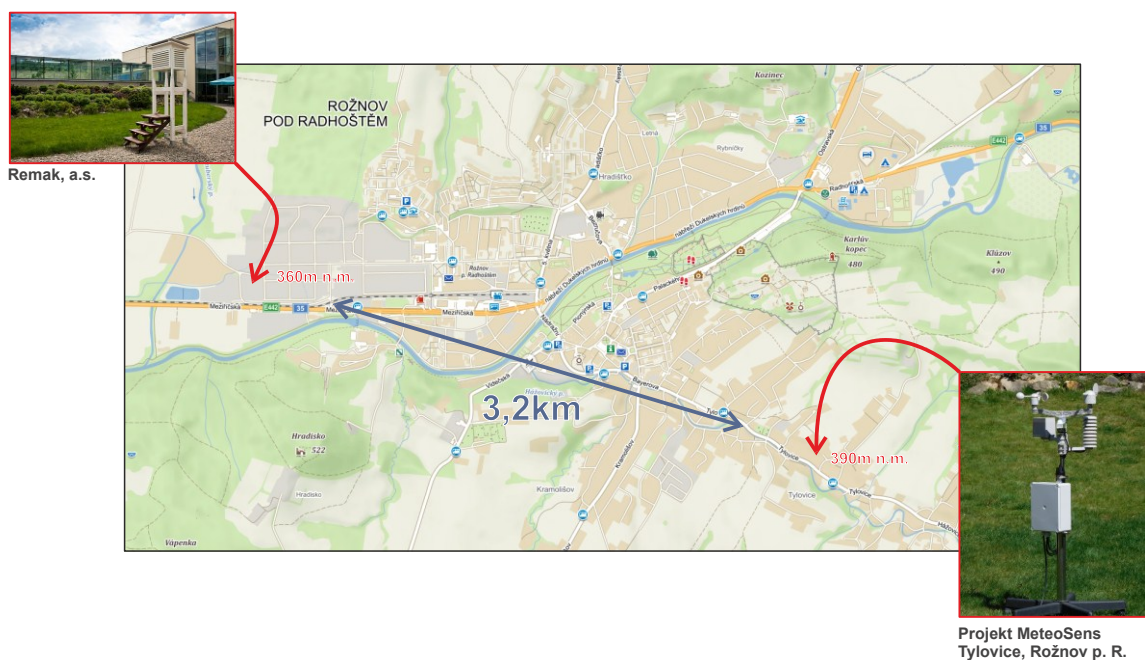
Z obr. 31 je patrná podoba přijímací jednotky. Jednotka byla navržena pro napájení síťového zdroje napětí nebo USB. Její odběr činí kolem 180mA, což znemožňuje provoz z baterie. Proto, aby mohl být použit provoz z baterie, musel by se dodatečně naprogramovat úsporný režim pro WI-FI část, úsporný režim pro displej (vypínání podsvitu) a také snížit frekvenci aktualizovaných hodnot. Nicméně protože přijímací jednotka bude umístěna v dosahu elektrické sítě, nebude potřeba její spotřebu optimalizovat, což také zajistí lepší spolehlivost.

Použitý displej disponuje také kapacitní dotykovou vrstvou, která je v tomto projektu neaktivní. Její nezprovoznění bylo zapříčiněno kvůli časové nedostupnosti, nicméně se jedná o kvalitní potenciál pro budoucí update.

Jednotka také umožňuje zobrazení aktuálního času, což je zajištěno funkcí `requestTime()`. Tato funkce, která je obsažena v knihovnách „MySensors“ a umožňuje posílání času z kontroléru do uzlu. Čas v kontroléru je shodný s časem v operačním systému, jenž je zpravidla aktualizován přes NTP server. Tzn., že čas zobrazený na displeji je poměrně přesný. Po zavolání funkce `requestTime()` dostáváme hodnotu v proměnné typu `uint32`. Tato hodnota představuje počet uplynulých sekund od data 1.1.1970. Hodnota je dále upravena na aktuální datum a čas speciálním algoritmem.

5 Ověření funkce a záznamy měření

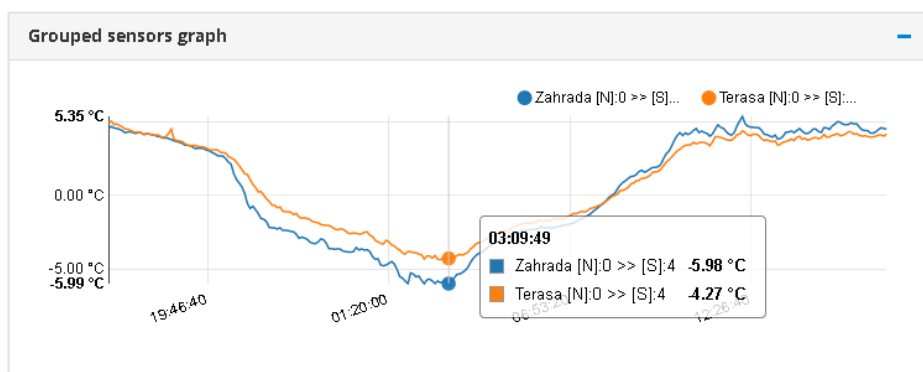
V rámci zadání diplomové práce bylo provedeno měření pro ověření správné funkce. Základní ověření lze provést pomocí rtuťového teploměru, což nám zaručí, že senzor teploty je funkční. Ale neověří stanici jako celek, v delším časovém rozmezí. Zvlášť největší problém meteostanic, neboli vliv přímého slunečního záření na změřenou teplotu. Proto je vhodné srovnat výsledky měření s nejbližší možnou profesionální meteostanicí. Meteostanice která připadá v úvahu a její data jsou veřejná, ale není ve vlastnictví ČHMU, nýbrž soukromé firmy Remak a.s, která se zabývá vzduchotechnikou. Provedení meteostanice je zřejmé z obr. 32, jedná se tedy o meteorologickou budku, ve které jsou senzory od výrobce Comet-Systems. Přesné parametry těchto senzorů nejsou známy, nicméně firma se na senzory přímo specializuje, tudíž jejich předpokládaná kvalita bude velmi dobrá. Vzdálenost mezi oběma stanicemi činí 3,2 km, rozdíl v nadmořské výšce 30m, dalším negativním faktorem je umístění v průmyslovém areálu, kde zdejší podniky provozují nepřetržitou výrobu. Tudíž změřená teplota v průmyslovém areálu bude vždy o něco větší, než v příměstské části Tylovice. Veličiny relativní vlhkosti a atmosférického tlaku by měly být více srovnatelné.



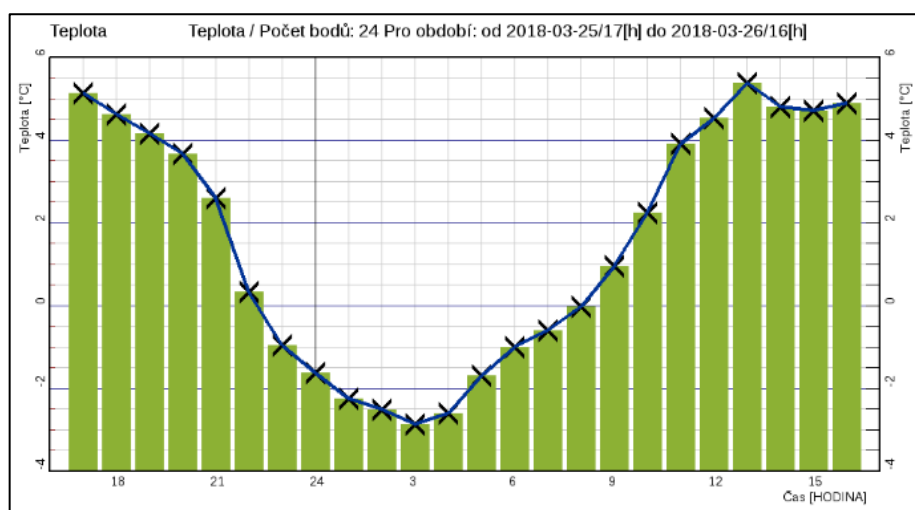
Obr. 32 Mapa umístění meteostanic, vzdálenost a výškový rozdíl

5.1 Měření teploty

Měření teploty bylo provedeno v časovém rozmezí 25.3.2018 16:40 – 26.3.2018 16:40. Stav oblačnosti téhož dne: zataženo, beze srážek. [10]



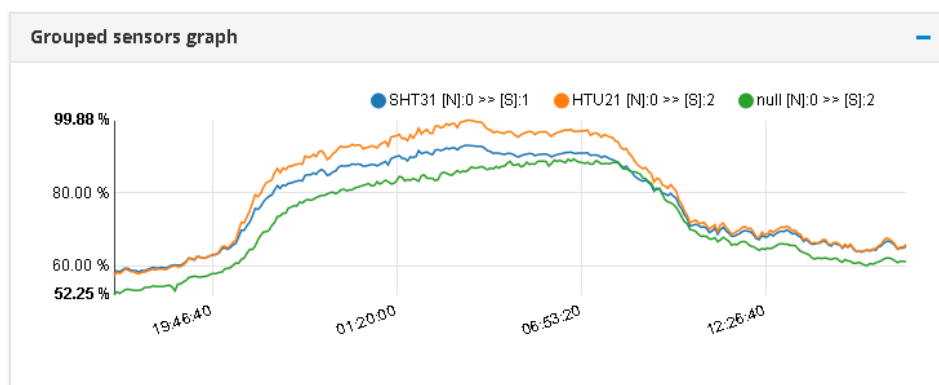
Obr. 33 Průběh teploty AVG – průměr všech čidel.



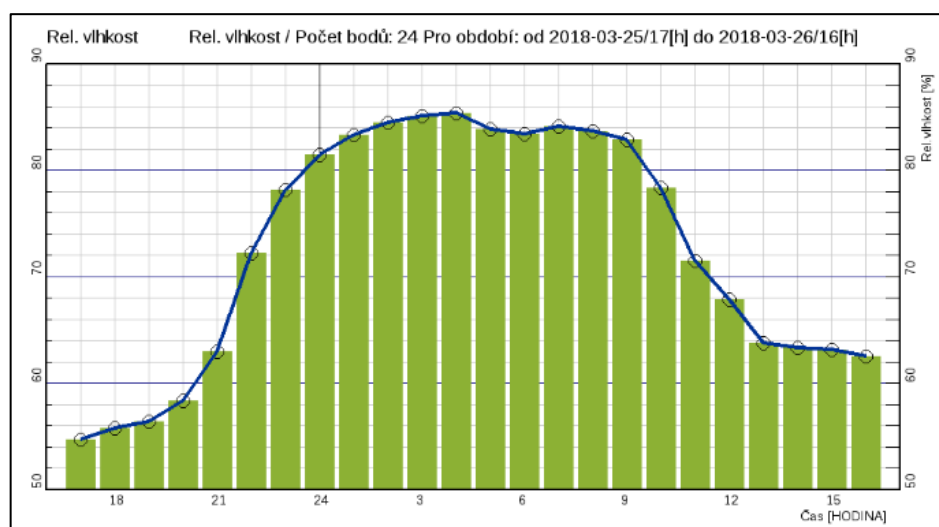
Obr. 34 Průběh teploty meteostanice Remak, a.s.

5.2 Měření vlhkosti

Měření vlhkosti bylo provedeno v časovém rozmezí 25.3.2018 16:40 – 26.3.2018 16:40. Stav oblačnosti téhož dne: zataženo, beze srážek. [10]



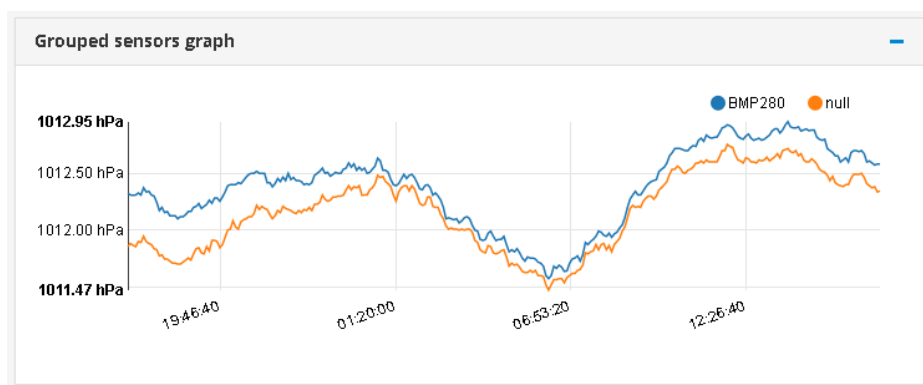
Obr. 35 Průběh relativní vlhkosti, porovnání čidel SHT31, HTU21 zahrada + čidlo HTU21 - krytá terasa



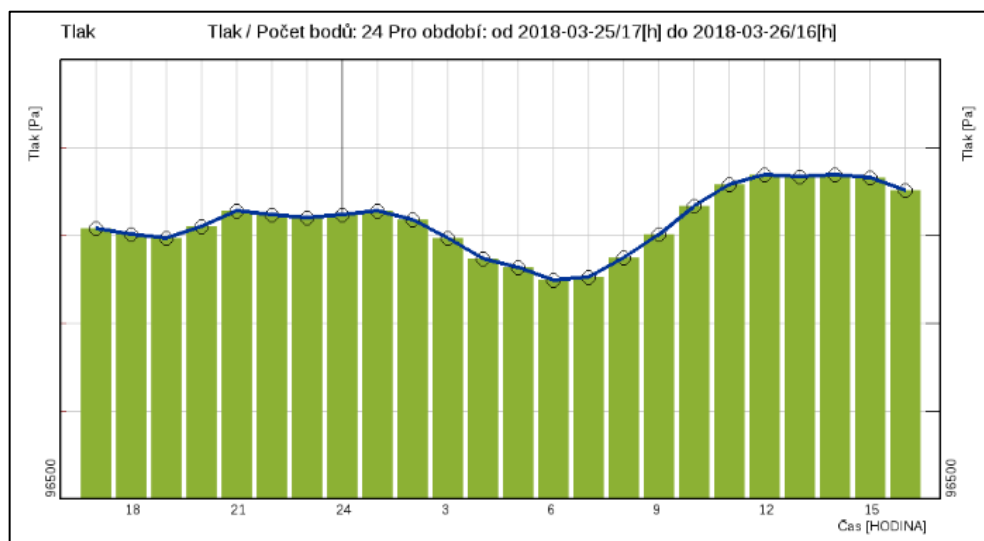
Obr. 36 Průběh relativní vlhkosti meteorostanice Remak, a.s.

5.3 Měření atmosférického tlaku

Měření atmosférického tlaku bylo provedeno v časovém rozmezí 25.3.2018 16:40 – 26.3.2018 16:40. Stav oblačnosti téhož dne: zataženo, beze srážek. [10]



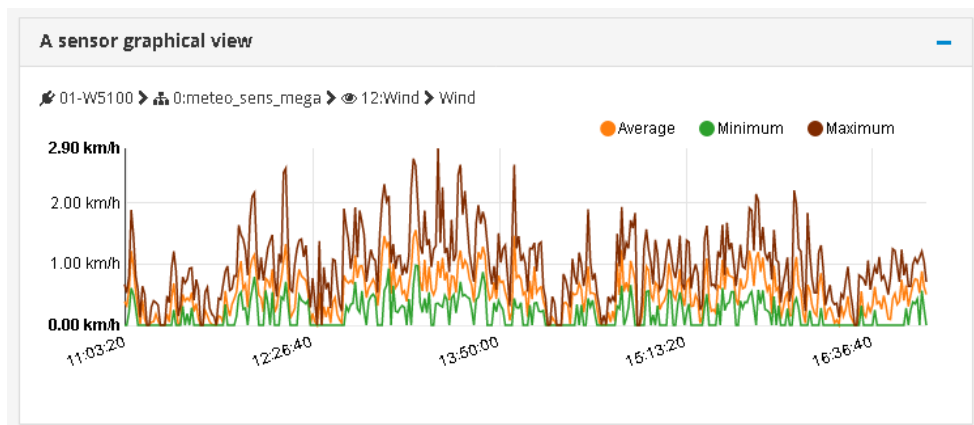
Obr. 37 Průběh atmo. tlaku, porovnání čidel BMP280-zahrada vs. BMP280-krytá terasa



Obr. 38 Průběh atmo. tlaku meteorostanice Remak, a.s.

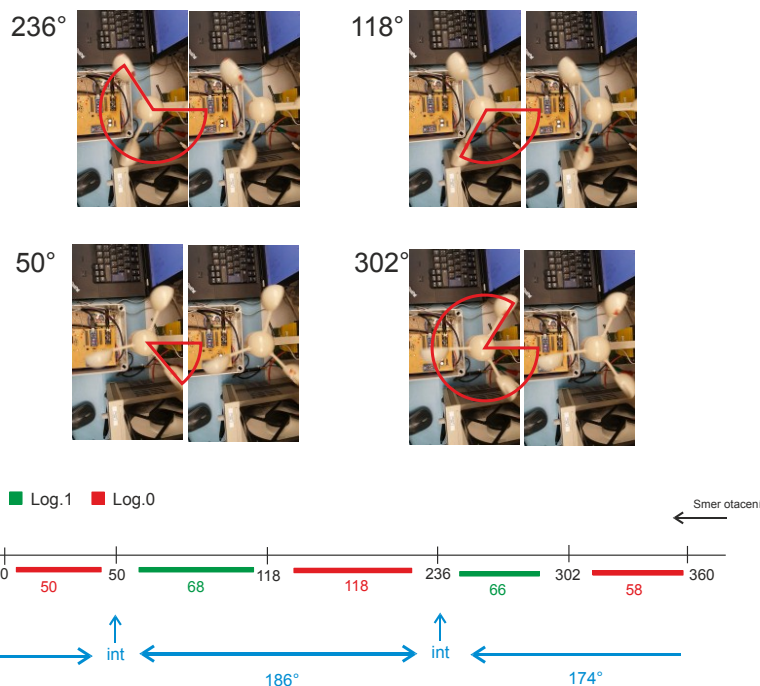
5.4 Měření rychlosti větru

Měření rychlosti větru bylo provedeno v časovém rozmezí 25.3.2018 10:40 – 16:40. Stav oblačnosti téhož dne: zataženo, beze srážek. Protože při měření větru má intenzita nárazový charakter, je vhodné zvolit v nastavení grafu volbu: Minimum, Maximum, Průměr. Po vykreslení grafu vidíme v daném časovém okamžiku hodnoty min, max, average, čímž můžeme snadno vidět hodnotu nárazového větru i průměrnou hodnotu rychlosti větru.

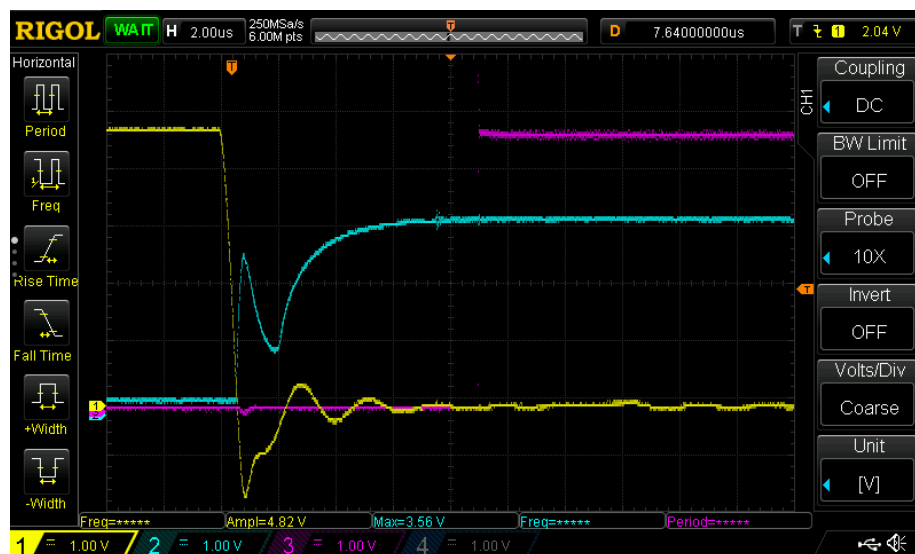


Obr. 39 Graf rychlosti větru (režim: Min,Max,Avg)

Ověření správné výstupní hodnoty rychlosti větru bylo provedeno pomocí kamery s vysokou snímkovací frekvencí (120fps). Vrtule anemometru je roztáčena konstantní rychlostí pomocí ventilátoru. Kamera snímá z vrchní strany anemometru. Při každém otočení o 180° nastane přerušení v MCU. Pokud porovnáme čas mezi jednotlivými přerušeními a čas zaznamenaný na kameře při zpomaleném záběru, můžeme tímto způsobem jednoduše ověřit správnost výstupních dat. Pokud tedy známe čas a dráhu, můžeme snadno vypočítat rychlost za sekundu.



Obr. 40 Výstupní logické úrovně v závislosti na úhlu natočení vrtule



Obr. 41 Průběh napětí na výstupu anemometru

Žlutý průběh – výstupní napětí, Růžový průběh – reakce přerušovacího systému

Modrý průběh – napětí na výstupu schmit. log obvodu, který byl v nové revizi desky vyřazen.

Graf na obr. 41 ukazuje průběh výstupního napětí na anemometru při sestupné hraně, která aktivuje přerušování. Je patrné, že zde dochází k určitému překmitu vlivem indukčnosti přívodního kabelu, nicméně zde je nejdůležitější parametr rychlost reakce přerušovacího systému, jenž činí 8 μ s. Tento parametr je dostatečný vzhledem k měřené periodě maximálně v desítkách milisekund.

5.5 Shrnutí výsledků

Ze všech uvedených grafů, tj. obr. 33 – 38 lze vyvodit, že trend veličin v čase je velmi podobný, i když absolutní hodnoty jsou jiné z důvodu vzdálenosti a umístění v jiné části města - průmyslovém areálu.

Např. z teplotních grafů na obr. 33, 34 lze vyčíst téměř konstantní tepelný rozdíl 3° C, dále pak z grafů vlhkosti lze vyčíst rozdíl v jednotkách procent, jako věrohodnější údaj je považovaná hodnota z čidla SHT31. Průběh tlaku v případě stanice firmy Remak je udáván v absolutních hodnotách, a není řádně označena osa Y, proto je srovnání nemožné.

Ověření správnosti údaje měření rychlostí větru bylo ověřeno pouze v laboratorních podmínkách. Věrohodnější výsledek by přinesl venkovní test s vozidlem vybavené GPS přijímačem pro odečítání aktuální rychlosti. Test by vyžadoval absolutní bezvětří, a za konstantní rychlosti vozidla by se ověřila správnost výstupních dat při větších rychlostech větru.

Závěr

Zadaným cílem této diplomové práce bylo rozebrat principy fyzikálních senzorů a vybrané konkrétní typy otestovat a vytvořit zařízení pro sběr dat fyzikálních veličin, což bylo splněno. Mým osobním cílem bylo vytvořit zařízení, které bude fungovat a bude bezporuchové a bezproblémové. Tohle je konstatováno s nadsázkou, v praxi to vidím tak, že i po odevzdání práce, a pevné instalaci, se bude zařízení nadále aktualizovat drobnými změnami, zejména část pro spínání ventilátoru cirkulace vzduchu radiačním krytem. Aktuální verze bude vždy dostupná na službě „GitHub“ a stažitelná pomocí odkazu:

https://github.com/hussit93/MeteoSens

Při testování jednotlivých senzorů se ukázalo že, některé levnější typy senzorů nejsou z výroby dostatečně zkaličované a je potřeba provést softwarové úpravy. V případě senzoru HTU21 je tato odchylka přijatelná a neodporuje údajům z datasheetu. V případě tlakoměru BMP280 je tato odchylka teploty výrazná, a je potřeba ji upravit.

Zařízení bylo navrženo tak, aby bylo po odladění softwaru již nepotřebovalo výraznou obsluhu. Zejména použití napájení z rozvodné sítě oproti napájení z baterií, tuhle vlastnost na první pohled zajistí. Nicméně systém je obecně vždy tvořen větším počtem částí, z níž pokud, jedna část selže, zařízení se stane nefunkčním. V tomto konkrétním případě je systém myšlen hlavně konkrétním přenosovým řetězcem, který byl v tomto projektu použit.

Jednotka sběru dat -> „Powerline“ -> bezdrátový síťový most -> „Switch“ -> „Server-Controler“.

V případě nežádoucí situace tj. výpadek elektrické sítě a následném obnovení provozu, se může stát že, jedna část z přenosového řetězce nenaběhne korektně, tím pádem je systém nefunkční a potřebuje zásah obsluhy, např. Restart síťového mostu nebo powerline adaptéru.

Z toho usuzuji, že do budoucna budu uvažovat o zálohování pomocí baterie, což v aktuální době znamená nepřiměřené vysoké náklady, vzhledem k použití jako informační zařízení. Ale v budoucí době po rozšíření systému o zabezpečovací prvky, bude bateriová záloha více předmětná záležitost.

V systému sítě „MySensors“ jsem našel zalíbení, proto se bude domácí projekt nadále rozšiřovat zejména v oblasti spínání venkovních světel a zabezpečení domu – požární senzory, pohybové senzory, okenní/dveřní senzory atd.

Přínos této práce vidím v otestování vybraných senzorů, a zhodnocení jejich výhod a nevýhod v použití v praxi. Dále pak zprovoznění systému sběru a ukládání dat fyzikálních veličin, který by měl být dostatečně stabilní v delším časovém rozmezí.

Stabilita také úzce souvisí s použitou verzí knihoven „MySensors“ a verzí „MyController“, jak už jsem se dávno přesvědčil, nejnovější software neznamená vždy nejstabilnější. Proto v tomto projektu byly vybrány tyto verze jako nejstabilnější:

- MySensors v2.1.1
- MyController v1.1.0 Final

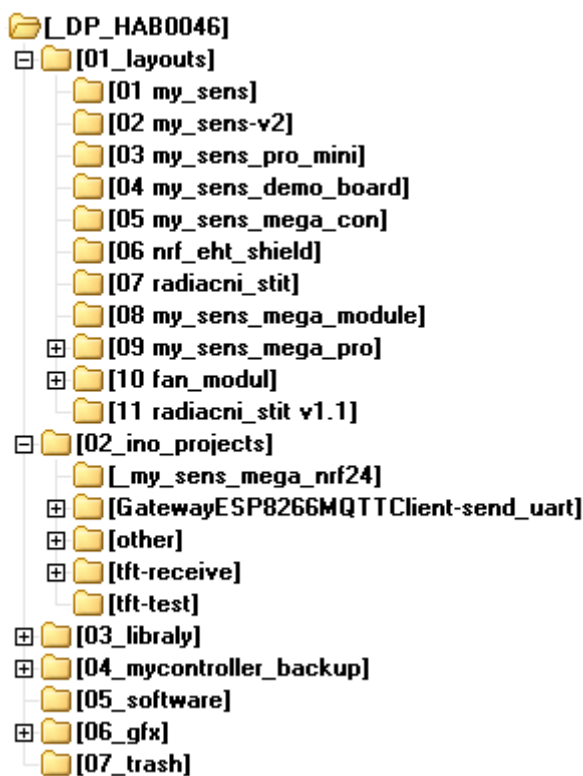
Použitá literatura

- [1] HEROUT, Pavel. *Učebnice jazyka C*. 3. upravené vydání, České Budějovice: Kopp, 1994. ISBN 80-85828-21-9.
- [2] MySensors web/forum [online] [cit. 2018-04-10]
Dostupné z WWW: <https://www.mysensors.org/>
- [3] Měření teploty [online] [cit. 2018-04-10]
Dostupné z WWW: https://www.wikiskripta.eu/w/Měření_teploty
- [4] Atmosférický tlak [online] [cit. 2018-04-10]
Dostupné z WWW: https://cs.wikipedia.org/wiki/Atmosférický_tlak
- [5] Fotodioda [online] [cit. 2018-04-10]
Dostupné z WWW: <http://fyzika.jreichl.com/main.article/view/275-fotodioda>
- [6] Datasheet SHT31 [online], [cit. 2018-04-10], Dostupné z WWW:
https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/2_Humidity_Sensors/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital.pdf
- [7] Datasheet HTU21 [online], [cit. 2018-04-10], Dostupné z WWW:
http://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FHPC199_6%7FA6%7Fpdf%7FEnglish%7FENG_DS_HPC199_6_A6.pdf%7FCAT-HSC0004
- [8] Datasheet BMP280 [online], [cit. 2018-04-10], Dostupné z WWW:
<https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>
- [9] Datasheet BH1750 [online], [cit. 2018-04-10], Dostupné z WWW:
<http://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>
- [10] Počasí Remak, Rožnov p. R. [online] [cit. 2018-04-10]
<http://www.remak.eu/cs/pocasi-v-roznove-pod-radhostem>

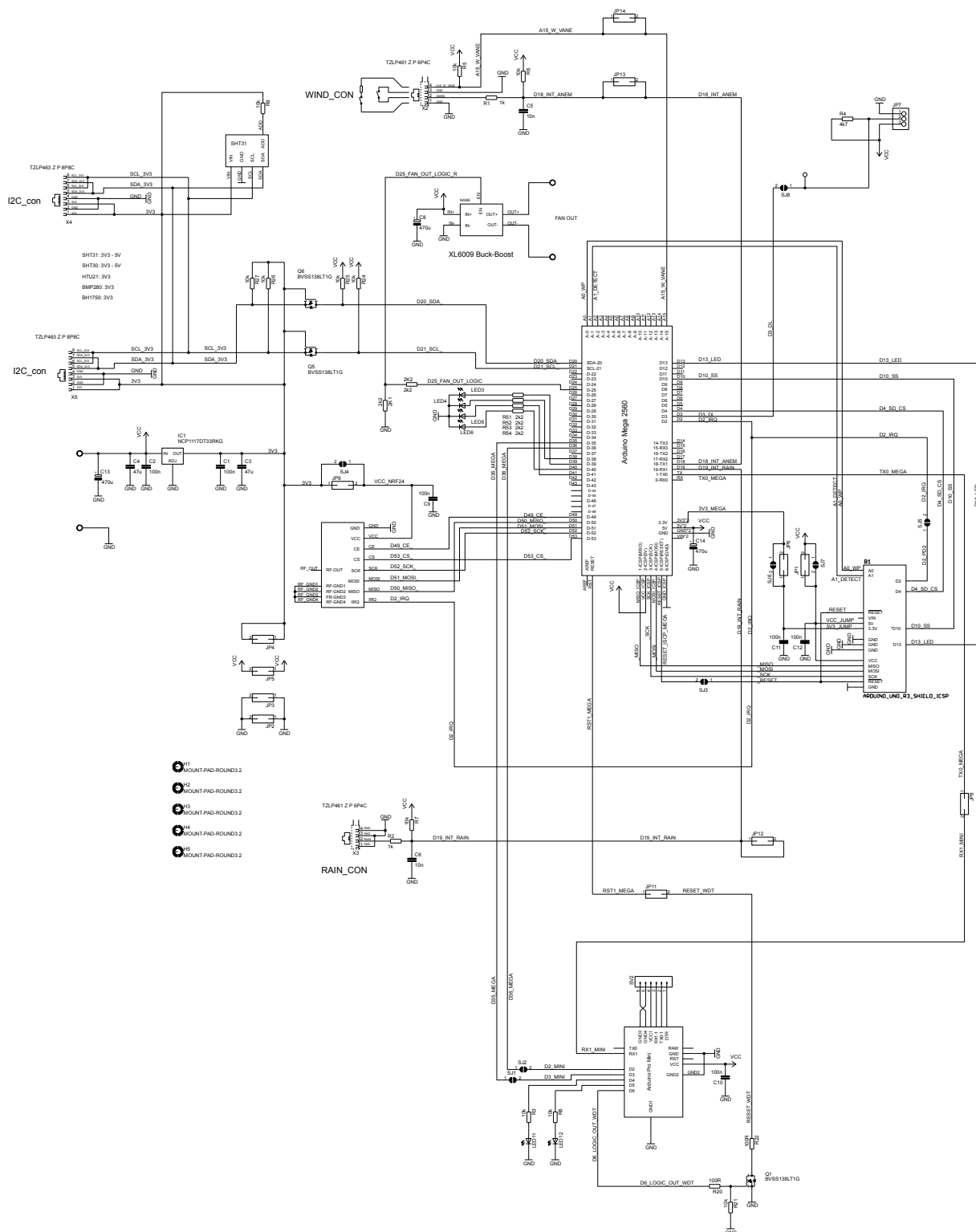
Seznam příloh

Příloha A:	Obvodové schéma - Jednotky sběru dat.....	II
Příloha B:	Foto	III
Příloha C:	Programové vybavení.....	IV
Příloha D:	Návrh plošného spoje	(součástí CD)

Adresářová struktura přiloženého CD:



Příloha A: Obvodové schéma



Příloha B: Foto celkové



Celkový projekt obsažen na CD, zde pouze výňatek důležitějších částí

Smyčka Loop()

```
void loop()
{
    timer_min();
    timer_hod();
    currentMillis = millis();
    if (currentMillis - previousMillis_sec >= interval_sec) {
        previousMillis_sec = currentMillis;
        id++;
        Serial.println(" ");
        Serial.println(" ");
        Serial.print("id: ");
        Serial.print(id);
        Serial.println(" -----");
        if(id < -1){
            boot_led_state = !boot_led_state;
            digitalWrite(40, boot_led_state);
        }
        if(id > 0){
            digitalWrite(41, HIGH);
            digitalWrite(38, LOW); //Stby LED
            wind_speed_send();
            wind_vane_send();
            uart_check();
        }
        if(id == -9){
            if(request_need == 1){
                if(rain_imp_cnt == 0){
                    Serial.print("request:");
                    request(RAIN_CHILD, V_VARS5);
                }
            }
            request_need = 0;
        }
        if(id == -7){
            sht31_send();
        }
        if(id == -4){
            htu21_send();
        }
        if(id == -3){
            bmp280_send();
        }
        if(id == -2){
            bh1750_send();
        }
        if(id == -1){
            avg_get();
            digitalWrite(40, LOW); // Boot LED
            digitalWrite(38, HIGH); //Stby LED
            interval_sec = 3000;
        }
        if(id == 1){
            Serial.print(" - Fan_state: ");
        }
    }
}
```

```

        Serial.println(fan_state);
        if (rain_simulation == 1){
            rain_click();
        }
    }
    if(id == 2){
        sht31_send();
    }
    if(id == 4){
        htu21_send();
    }
    if(id == 5){
        bmp280_send();
    }
    if(id == 6){
        avg_get();
    }
    if(id == 7){
        bh1750_send();
    }
    if(id == 8){
        uptime_send();
    }
    if(id == 9){
        rain_send();
        id = 0;
    }
    digitalWrite(41, LOW);
    digitalWrite(38, HIGH);
}
}

```

Funkce sht31_send()

```
void sht31_send(){
    t_sht_raw = sht31.readTemperature();
    h_sht_raw = sht31.readHumidity();
    Serial.println("      SHT31_inboard:");
    t_sht_raw_2 = sht31_2.readTemperature();
    send(sht2_t_Msg.set(t_sht_raw_2,2));
    h_sht_raw_2 = sht31_2.readHumidity();
    send(sht2_h_Msg.set(h_sht_raw_2, 1));
    Serial.println(" ");
    delay(delay_time);
    if(t_sht_raw != 0){
        if(t_sht_raw < horni_t_limit && t_sht_raw > dolni_t_limit){
            t_sht = t_sht_raw;
            Serial.println("      SHT31:");
            send(sht_t_Msg.set(t_sht,2));
            delay(delay_time);
        }
    }
    if(h_sht_raw != 0){
        if(h_sht_raw < 99.9 && h_sht_raw > 4.5){
            h_sht = h_sht_raw;
            send(sht_h_Msg.set(h_sht, 1));
            h_sht = 0;
            delay(delay_time);
        }
    }
    if(t_sht_raw && h_sht_raw){
        dP=(dewPointFast(t_sht_raw, h_sht_raw));
        send(dp_msg.set(dP,1));
        delay(delay_time);
        Serial.println(" ");
    }
    #ifdef SHT31_SHOW
    Serial.print("Temp SHT31: ");
    Serial.println(t_sht_raw);
    Serial.print("Humidity SHT31: ");
    Serial.println(h_sht_raw);
    Serial.print("Dew Point: ");
    Serial.println(dP);
    Serial.println(" ");
    Serial.print("Temp SHT31_2: ");
    Serial.println(t_sht_raw_2);
    Serial.print("Humidity SHT31_2: ");
    Serial.println(h_sht_raw_2);
    #endif /* MACRO */
}
```